



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DEL TÍTULO DE

INGENIERO EN MECATRÓNICA

TEMA:

**“ESTUDIO COMPARATIVO DE DOS MÉTODOS DE REDUCCIÓN DE
DIMENSIONES EN LA IDENTIFICACIÓN DE PATRONES DE MOVIMIENTO
EN SEÑALES ELECTROMIOGRÁFICAS”.**

AUTOR:

GABRIEL ELÍAS PAREDES MENDOZA

DIRECTOR:

MSC. IVÁN IGLESIAS NAVARRO

IBARRA, 2019



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN

A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	DE	1312420779	
APELLIDOS Y NOMBRES:	Y	Paredes Mendoza Gabriel Elías	
DIRECCIÓN:		Ibarra	
EMAIL:		geparedesm@utn.edu.ec	
TELÉFONO FIJO:		052920549	TELÉFONO MÓVIL: 0968594103

DATOS DE LA OBRA	
TÍTULO:	ESTUDIO COMPARATIVO DE DOS MÉTODOS DE REDUCCIÓN DE DIMENSIONES EN LA IDENTIFICACIÓN DE PATRONES DE MOVIMIENTO EN SEÑALES ELECTROMIOGRÁFICAS
AUTOR (ES):	Paredes Mendoza Gabriel Elías
FECHA: DD/MM/AAAA	29/07/2019
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	PREGRADO
TÍTULO POR EL QUE OPTA:	Ingeniero en Mecatrónica
ASESOR /DIRECTOR:	Ing. Iván Iglesias Navarro

2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 29 días del mes de julio de 2019

EL AUTOR:



Firma

Gabriel Elías Paredes Mendoza



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADA

CERTIFICACIÓN

En calidad del director del trabajo de grado “ESTUDIO COMPARATIVO DE DOS MÉTODOS DE REDUCCIÓN DE DIMENSIONES EN LA IDENTIFICACIÓN DE PATRONES DE MOVIMIENTO EN SEÑALES ELECTROMIOGRÁFICAS”, presentado por el egresado GABRIEL ELÍAS PAREDES MENDOZA, para optar por el título de Ingeniero en Mecatrónica, certifico que el mencionado proyecto fue realizado bajo mi dirección.

Ibarra, a los 29 días del mes de julio de 2019

A handwritten signature in blue ink, consisting of a stylized 'I' and 'G'.

MSc. Iván Iglesias
DIRECTOR DE TESIS

Agradecimiento

Todos los sentimientos encontrados al terminar esta etapa de mi vida son indescriptibles, cada una de las personas que han sido parte de mi formación personal y académica tienen un lugar muy importante dentro de mi corazón.

Agradezco primeramente a mis padres Edil mina Mendoza y Bolívar Paredes que sé que sienten orgullosos al verme cumplir esta importante meta en mi vida, por todo el apoyo brindado, su paciencia y amor ya que sin ellos nada de esto hubiese sido posible, han sido la inspiración de todos mis sueños, me han enseñado a combatir los obstáculos que se presentan con valentía y rigor, me han cuidado en cada uno de pasos que he dado, simplemente me enseñaron todo lo que una persona necesita saber para defenderse, salir adelante, levantar de una caída y los más importante que una persona puede tener en su vida que es la humildad. No existen palabras concretas que puedan describir todo lo que significan para mí. Todas aquellas personas que formaron parte de mi vida y de las cuales de una u otra manera siempre aprendí algo, fue muy gratificante encontrarme en el camino con compañeros de clase con los cuales compartimos tareas, proyectos y más que nada risas. Agradezco a mis profesores que me enseñaron todo lo necesario para realizar este trabajo y sobre todo a mi tutor el cual más que un tutor supo ganarse una amistad con la cual el trabajo fue mucho más gratificante y ameno.

Gracias a todos.

Dedicatoria

Dedico este proyecto a mis padres, quienes siempre estuvieron presente en cada uno de mis pasos durante el forjamiento de mis conocimientos y con su sacrificio permitieron que pueda culminar una meta más en mi vida, mismo que me ayudara en el desenvolvimiento de mi vida profesional.

Gabriel Elías Paredes Mendoza

Resumen

La reducción de dimensiones es una técnica muy utilizada en diferentes aplicaciones, en este estudio se la utilizará en las señales electromiográficas. Este es un proceso que se encarga de disminuir la información de la base de datos inicial, lo cual reduce el tiempo de procesamiento de información, posibilitando la futura implementación a nivel de microprocesador y la ejecución de otras tareas en los tiempos liberados.

Existen diversos métodos que ayudan a reducir dimensiones, entre ellos se encuentran: Análisis de componente principal (ACP), Análisis Discriminante Lineal (ADL), Análisis de Componentes Independientes (ACI), Método de Regresión Lineal Múltiple (MRLM). A; encontrar diversos enfoques aparece la necesidad de verificar su desempeño computacional y capacidad de acierto en clasificación, para lo cual este trabajo realiza un estudio comparativo de los métodos más populares citados anteriormente.

Cada uno de los algoritmos ha sido implementado en *Matlab*. La base de datos utilizada para la prueba de funcionamiento de los algoritmos contiene 8000 valores, distribuidos en 4 movimientos y 11 características. Los resultados obtenidos se han evaluado considerando el porcentaje de acierto de clasificación y tiempo de ejecución.

Al finalizar la evaluación se obtuvo como resultado que el método de reducción de dimensiones *ACP* arroja como resultado un mayor acierto de clasificación y menor tiempo que ejecución que el *MRLM*.

Abstract

The reduction of dimensions is a very useful technique in different applications, in this study it is used in electromyographic signals. This is a process that is responsible for reducing the information in the initial database, which reduces the time of information processing, enabling future implementation at the microprocessor level and the execution of other tasks in the released times.

There are several methods that help reduce the dimensions, among them are Principal Component Analysis (ACP), Linear Discriminant Analysis (ADL), Independent Component Analysis (ACI), Multiple Linear Regression Method (MRLM). A; find in the same place the need to verify their computational performance and the ability to succeed in the classification, for which this work makes a comparative study of the most popular methods mentioned above.

Each of the algorithms has been implemented in MatLab. The data base for the algorithm function test contains 8000 values, distributed in 4 movements and 11 characteristics. The results have been evaluated the percentage of classification success and execution time.

At the end of the evaluation, the ACP dimension reduction method resulted in a greater classification success and less time than the execution of the MRLM.

Índice general

Introducción	1
Motivación	1
Objetivos	2
Objetivo General	2
Objetivos Específicos.....	2
Alcance	2
Justificación	2
1. Marco Teórico.....	3
1.1. Sistema neuromuscular.....	3
1.2. Fibras musculares y Fibras nerviosas	3
1.3. Unidad motora	4
1.4. Osteología del miembro superior	5
1.5. Anatomía del brazo.....	6
1.6. Movimientos.....	6
1.7. Señales Electromiográficas.....	7
1.8.1. Adquisición de señales electromiográficas	7
1.8.2. Características de las señales electromiográficas.....	8
1.8. Reducción de dimensiones	11
1.9.1. Análisis de componente principal (ACP).....	11
1.9.2. Análisis Discriminante Lineal (ADL)	12
1.9.3. Análisis de Componentes Independientes (ACI)	13
1.9.4. Método de Regresión Lineal Múltiple (MRLM).....	15
2. Marco Metodológico.....	17
2.1. Selección de métodos de reducción de dimensiones	17
2.2. Ingreso de datos	18
2.3. Modificación de la Base de Datos	18
2.2.1. Normalización	18
2.2.2. Matriz de entrenamiento.....	18
2.2.3. Matriz de prueba.....	18
2.4. Métodos de Reducción de Dimensiones.....	20
2.3.1. Análisis de Componente Principal	20
2.3.2. Método de regresión lineal múltiple	21
2.5. Obtención bases de datos con dimensión reducida	22
2.6. Métodos de clasificación	23

3. Análisis y Resultado	25
Algoritmos	33
Reducción de dimensiones.....	33
Clasificación	39

Índice de figuras

Figura 1.1: Sistema Nervioso y Muscular.....	3
Figura 1.2: Partes de la neurona.....	4
Figura 1.3: Unidad Motora	5
Figura 1.4: Complejo articular del miembro superior del ser humano.	5
Figura 1.5: Rotación del brazo con referencia al antebrazo.....	6
Figura 1.6: Movimiento del brazo.....	7
Figura 1.8: Electromiografía de superficie cruda de tres contracciones del bíceps..	8
Figura 2.1 Diagramas de flujo de Funciones de Normalización, Matriz Entrenamiento y Matriz Prueba	19
Figura 2.2 Diagrama de flujo del Análisis de Componente Independiente.	20
Figura 2.3 Pagina 1 Diagrama de Flujo de Método de Regresión Lineal Múltiple	21
Figura 2.4 Pagina 2 Diagrama de Flujo de Método de Regresión Lineal Múltiple	22
Figura 2.5 Diagrama de Flujo de las Bases de Datos de dimensión reducida	22
Figura 2.6 Diagrama de Flujo de Clasificación de Características	23
Figura 2.7 Diagrama de flujo general	24
Figura 3.1 Acierto y error de la clasificación de la base de datos de once características.	25
Figura 3.2 Acierto y error de la clasificación de la base de datos de dimensión reducida de tres características por método de <i>ACP</i>	26
Figura 3.3 Acierto y error de la clasificación de la base de datos de dimensión reducida de tres características por <i>MRLM</i>	27
Figura 3.4 Comparativa de porcentaje de acierto de clasificación de base de datos de once dimensiones, dimensión reducida por <i>ACP</i> y dimensión reducida de <i>MRLM</i>	27

Índice de Tablas

Tabla 2.1 Matriz de características de un movimiento.	18
Tabla 2.2 Matriz de características normalizada.....	19
Tabla 3.1 Métodos de reducciones con sus tres principales características.....	25
Tabla 3.2 Tiempo de procesamiento	28
Tabla 3.3 Característica del computador.....	28

Introducción

Este trabajo de grado ha sido realizado con el *Grupo de Investigación en Sistemas Inteligentes de la Universidad Técnica del Norte (GISI-UTN)*.

Motivación

En el transcurso del tiempo y con la evolución de la tecnología se ha dado mucha importancia a las investigaciones enfocadas a realizar interfaces hombre-máquina, teniendo como objetivo principal ayudar a personas con discapacidad o ejecutar tareas sin contacto directo por parte del operador [1]. Una de las formas más utilizadas para conseguir esa meta es la captación de señales electromiografías (EMG) provenientes de los músculos que están asociadas cualitativamente con el tipo de movimiento producido [2].

Las señales EMG se han convertido en un elemento importante en reconocimiento de patrones, control de dispositivos, rehabilitación, entre otros campos debido a la facilidad con la que se pueden adquirir de forma no invasiva y sin peligro para el individuo [3]. Siendo estas aplicaciones evaluadas por su desempeño mediante la ejecución de movimientos prácticos, con la naturalidad de los ejecutados por la mano humana, la menor complejidad de operación como sistema y con el menor esfuerzo físico-mental por parte del paciente [4]. Estos parámetros son el resultado de un correcto preprocesamiento de las características de las señales EMG; siendo identificadas más de cuarenta características aleatorias, no estacionarias y no lineales al momento de tomar la señal, lo cual ocasiona bajo desempeño, alto costo y pérdida de tiempo en el procesamiento [3].

Una vez que se obtiene la información, existe un gran número de características que al momento de procesarla se requiere de hardware y tiempo suficiente para obtener buenos resultados, por lo cual es indicado aplicar un método de reducción de señales, que consiste en disminuir el número de variables a tratar. De esta manera se disminuye los requisitos mínimos de hardware, tiempo de procesamiento y aumenta el ahorro de energía y la rapidez de procesar la información [3].

La reducción de dimensiones es frecuentemente usada como una etapa de pre-proceso en el entrenamiento de sistemas, y consiste en escoger un subconjunto de variables, de tal manera, que el espacio de características quede óptimamente reducido de

acuerdo con un criterio de evaluación, cuyo fin es distinguir el subconjunto que representa mejor el espacio inicial de entrenamiento. Como cada característica que se incluye en el análisis, incrementa el costo y el tiempo de proceso de los sistemas [5] , hay una fuerte motivación para diseñar e implementar sistemas con pequeños conjuntos de características. Sin dejar de lado que, al mismo tiempo, hay una opuesta necesidad de incluir un conjunto suficiente de características para lograr un alto rendimiento [6].

Objetivos

Objetivo General

- Comparar dos métodos de reducción de dimensiones para la selección de características principales de la señal EMG.

Objetivos Específicos

- Seleccionar los algoritmos de reducción de dimensiones desde la literatura.
- Implementar los métodos de reducción de dimensiones a través de software matemático.
- Analizar los resultados en base al desempeño en la clasificación de señales EMG.

Alcance

En el presente estudio se realizará la implementación de dos métodos de reducción de dimensiones para la identificación de patrones de movimiento en la señal EMG, utilizando bases de datos de repositorios libres. Los resultados serán validados aplicando dos métodos de clasificación utilizados en investigaciones anteriores, seleccionados de bibliografía especializada.

Justificación

La idea parte de la necesidad de realizar un correcto control de los movimientos de las prótesis y dispositivo de rehabilitación diseñados en la carrera de *Ingeniería en Mecatrónica de la Universidad Técnica del Norte*. Por lo que en el *Grupo de Investigación de Sistemas Inteligentes* se propone un proyecto para dar cumplimiento a este propósito; planteándose como premisa que, para realizar un correcto control, lo primero es tener un pleno conocimiento de los patrones identificativos de la señal EMG que están presentes en cada movimiento.

Mediante los algoritmos que se implementarán se pretende reducir las dimensiones obtenidas de las señales electromiográficas y, de esta manera las características más importantes de la señal EMG reducirán el tiempo de procesamiento de los algoritmos de clasificación.

Capítulo 1

1. Marco Teórico

En el presente capítulo se analizarán aspectos básicos de anatomía y fisiología muscular que resultan necesarios conocer para un adecuado análisis de la señal electromiográfica (EMG). Inicialmente se expondrán una revisión del sistema neuromuscular, se explicará la estructura de la unidad motora, se analizará los movimientos que se realizan en torno a la articulación de la muñeca respecto al plano sagital. Además, se presentan los fundamentos de señales electromiográficas con sus respectivas características y los métodos de reducción de dimensiones.

1.1. Sistema neuromuscular

El sistema neuromuscular comprende el sistema nervioso, y el sistema musculoesquelético como se aprecia en la Figura 1.1, estos sistemas trabajan en cooperación principalmente para la producción de movimientos a partir de la recepción de un estímulo. El sistema nervioso facilita un trabajo armonioso entre el esqueleto, articulaciones y músculos para la generación de movimientos. Formado por el cerebro, la espina dorsal y los nervios, que son los encargados de captar y procesar rápidamente las señales ejerciendo control y coordinación sobre los demás órganos. El sistema muscular, compuesto por más de 600 músculos genera movimientos en el cuerpo siendo estos, de tipo voluntarios o involuntarios, contribuye además a proporcionar estabilidad. Se controla todo este sistema a través del sistema nervioso, aunque algunos músculos funcionan de manera autónoma [7], [8].

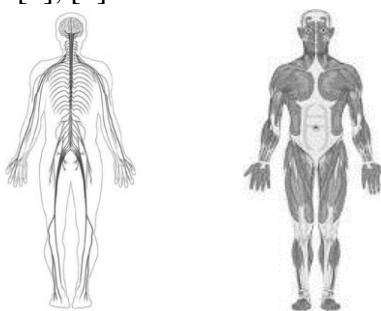


Figura 1.1: Sistema Nervioso y Muscular.

1.2. Fibras musculares y Fibras nerviosas

La fisiología muscular depende directamente de la excitabilidad de las fibras musculares ocasionadas por el control neural. Propiedades eléctricas del sarcolema (membrana citoplásmica de las células musculares) y su semipermeabilidad logran producir este fenómeno. El equilibrio iónico entre el interior y exterior de la célula muscular forma una diferencia de potencial en la membrana de la fibra muscular (aproximadamente entre -80mV y -90mV cuando no existe contracción). Esta diferencia de potencial se mantiene por procesos fisiológicos (bomba de iones) y se traduce en una

carga negativa intercelular respecto a la superficie externa. La activación de una célula alfa-motora de asta anterior (incluido por el sistema nervioso central o relejo) se convierte en la conducción de la excitación a través del nervio motor. Después de la liberación de sustancias transmisoras en las placas terminales motoras, un potencial de la placa terminal se forma en las fibras musculares inervadas por esta unidad motora. Las características de difusión de la membrana de la fibra muscular son ligeramente modificadas por iones de sodio que ingresan. Esto causa una repolarización de la membrana que es inmediatamente restaurada por un intercambio, la repolarización [9].

Gran parte de las células nerviosas tienen prolongaciones extensas (axones), lo que permite la conducción de impulsos eléctricos a lo largo de todo el cuerpo (

Figura 1.2). Los axones se estimulan por presión, calor, factores químicos o choques eléctricos para lograr producir un potencial de acción. Al existir estimulación que llega desde el medio ambiente, inicia una rápida sucesión de potenciales de acción o un tren de pulsos ya sean de baja o alta frecuencia. Dependiendo del tipo de nervios la velocidad de propagación varía entre 1m/s y 100m/s. Durante el cambio de potencial los iones de sodio penetran el axón y los iones de potasio salen del mismo, pero las concentraciones se restablecen al axón restablecen rápidamente. El potencial de acción se propaga a lo largo del axón debido al ingreso y egreso de estos iones [10].

Una fibra de musculo estirado está rodeada por una membrana delgada con un potencial de reposo cercano al de la fibra nerviosa; cuando un impulso llega a esta fibra, esta se contrae y propaga un potencial de acción. Los nervios y los músculos son conocidos como tejidos excitables debido a que en ellos se detectan fácilmente cambios de potencial eléctrico[10],[11], [12].

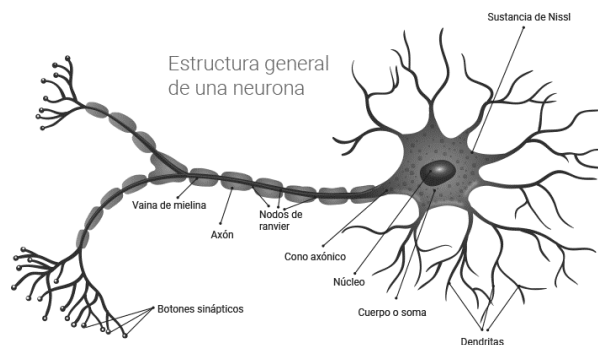


Figura 1.2: Partes de la neurona

1.3. Unidad motora

La unidad funcional más pequeña que describe el control del proceso de contracción muscular es llamada unidad motora (Figura 1.3), está compuesta por:

- Cuerpo celular y dendritas de una neurona motora.
- Las múltiples ramificaciones del axón.
- Las fibras musculares inervadas por el axón y sus ramas.

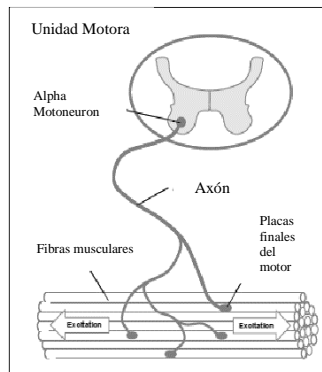


Figura 1.3: Unidad Motora

La unidad motora puede contener desde unas pocas fibras musculares (en el músculo laríngeo) a varios de cientos (en los músculos gemelos). Una regla general establece que mientras menor cantidad de fibras musculares sean inervadas por una misma neurona se conseguirá mayor precisión en el movimiento. Las fibras musculares que pertenecen a una misma unidad motora no se encuentran a distancias muy cercanas entre ellas; se encuentran esparcidos en un área pequeña del músculo y se mezclan con fibras de otras unidades motoras. El potencial de acción de unidad motora es el campo eléctrico generado por fibras musculares que pertenecen a una unidad motora. Comúnmente, las fibras musculares pertenecientes a una misma unidad motora se despolarizan y repolarizan de modo síncrono[9], [12], [13].

1.4. Osteología del miembro superior

En el cuerpo humano, los miembros superiores cada uno de las extremidades que se fijan a la parte superior del tronco. El miembro superior. El cingulo se encuentra formado por cuatro huesos: dos escápulas y dos clavículas; su función es la de conectar el miembro superior al esqueleto axial. Está compuesto por cuatro partes fácilmente distinguibles: mano, antebrazo, brazo y cintura escapular. En otras palabras, va desde el hombro hasta los dedos [14].

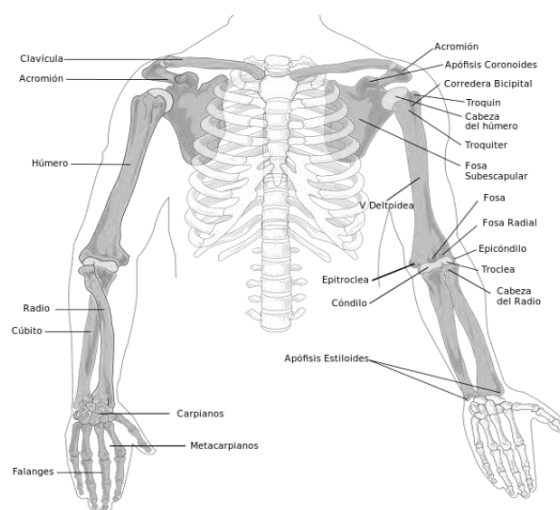


Figura 1.4: Complejo articular del miembro superior del ser humano.

1.5. Anatomía del brazo

El esqueleto del brazo consta de un solo hueso el húmero, en su parte diafisaria, es casi rectilíneo, algo torcido sobre su eje, irregularmente cilíndrico en su parte proximal y en su mitad distal afecta la forma de un prisma triangular [15].

La región anterior comprende tres músculos: en el plano superior, el musculo bíceps con sus dos porciones (corta y larga), y en un plano profundo el coracobraquial por arriba y el branquial anterior por abajo; la región posterior del brazo comprende solo el musculo tríceps branquial. El paquete vasculonervioso hace un recorrido longitudinal por el lado antero interno y el nervio radial rodea al húmero en forma de espiral profunda siguiendo el canal de torsión del tercio medio del hueso [15].

1.6. Movimientos

Anatómicamente el codo representa una sola articulación, en realidad no hay más que una sola cavidad articular; en cambio la fisiología nos permite distinguir dos funciones distintas:

La flexoextensión que precisa el concurso de dos articulaciones: humerocubital y humeroradial y la pronosupinación que pone en movimiento la articulación radiocubital proximal como se muestra en la Figura 1.5 [15].

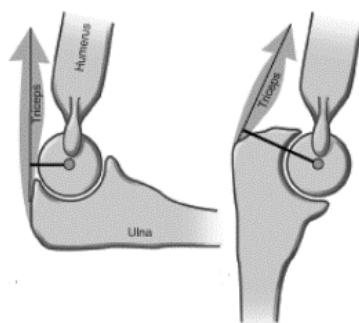


Figura 1.5: Rotación del brazo con referencia al antebrazo

El eje del antebrazo está situado en la prolongación del eje del brazo siendo ésta la posición de referencia; la extensión es el movimiento que lleva el antebrazo hacia atrás, la posición de referencia corresponde a la extensión completa, no existiendo amplitud de extensión del codo por definición (salvo ciertos sujetos con hiperlaxitud ligamentosa que pueden efectuar de 5° a 10° de hiperextensión del codo). La flexión es el movimiento que lleva el antebrazo hacia adelante aproximando la cara anterior del antebrazo a la cara anterior del brazo; la amplitud de flexión activa es de 145° y la de flexión pasiva es de 160°, la muñeca no llega a establecer contacto con el hombro. La pronosupinación sólo puede ser estudiada cuando el codo está en flexión de 90° y pegado al cuerpo; se considera posición intermedia cuando la dirección del pulgar es hacia arriba y la palma de la mano hacia dentro; la posición de supinación se realiza cuando la palma de la mano se dirige

hacia arriba con el pulgar hacia fuera y se encuentra situada en un plano horizontal y la amplitud de movimiento es de 90°. La posición de pronación se realiza cuando la palma de la mano mira hacia abajo con el pulgar hacia dentro, no alcanza a situarse en el plano horizontal y su amplitud de movimiento es de 85°. La posición funcional del codo es en flexión de 90° y prono-supinación [14] [15]..

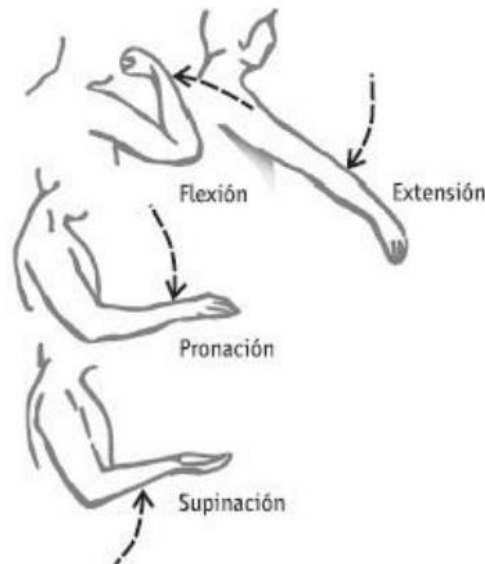


Figura 1.6: Movimiento del brazo.

1.7. Señales Electromiográficas

Las señales Electromiográficas, son señales eléctricas producidas por un músculo durante el proceso de contracción y relajación. Esta característica muscular se pretende aprovechar en el control por computadora para conseguir crear interfaces de comunicación entre el usuario y la máquina. Una interfaz de este tipo permitiría a cualquier usuario controlar sistemas informáticos y/o electrónicos mediante la contracción de determinados músculos. Un sistema de este tipo es especialmente interesante para individuos que padezcan algún tipo de parálisis que dificulte sus actividades y su interacción con el mundo que lo rodea, ofreciendo unas posibilidades funcionales hasta ahora inexistentes y mejorando su calidad de vida.

1.8.1. Adquisición de señales electromiográficas

Las señales electromiográficas, son adquisición típicamente mediante electrodos, ubicados sobre la piel. Estos han sido utilizados para el control de prótesis de miembros superiores desde 1984 [16]. Estas señales proveen información sobre la actividad neuromuscular que las origina [17].

1.8.2. Características de las señales electromiográficas

Una señal de electromiografía sin procesar que detecta los potenciales de acción de las unidades motoras superpuestas se denomina señal de EMG RAW. En el siguiente ejemplo (Figura 1.7), se muestra un registro de electromiografía de superficie cruda para tres contracciones del músculo bíceps.

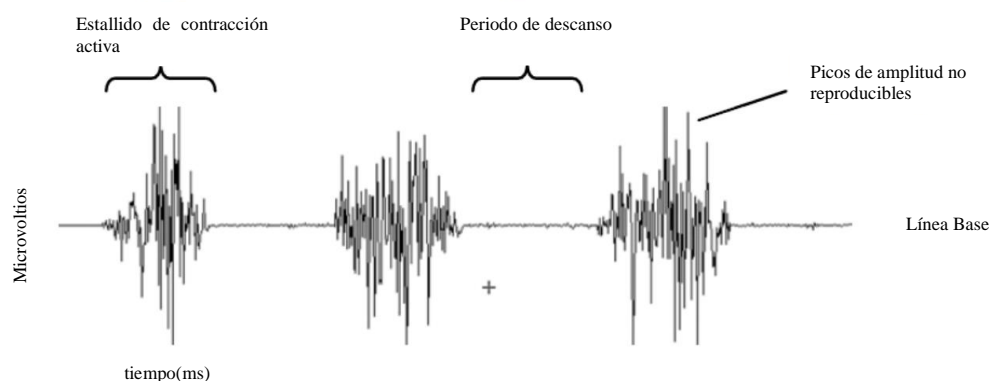


Figura 1.7: Electromiografía de superficie cruda de tres contracciones del bíceps

Cuando el músculo está relajado, se puede ver una línea base EMG más o menos libre de ruido. El ruido básico de EMG depende de muchos factores, especialmente de la calidad del amplificador EMG, el ruido ambiental y la calidad de detección dada. Suponiendo un rendimiento de amplificador de última generación y una preparación adecuada de la piel (desarrollado en los apartados siguientes), el ruido de referencia promedio no debe ser superior a 3-5mV, aunque el objetivo debe ser de 1 a 2mV. Adicionalmente, es importante controlar la calidad del electrodo de referencia de EMG, siendo un punto de control muy importante de cada medición de EMG. Por otra parte, hay que tener cuidado de no interpretar al ruido de interferencia o a los problemas dentro del aparato de detección como actividad de base "aumentada" o hipertono muscular [18].

Una electromiografía de superficie sin procesar puede oscilar entre $\pm 5000\text{mV}$, para atletas, siendo un tanto menor en personas no entrenadas. Típicamente el ancho de frecuencia oscila entre 6 y 500 Hz, mostrando la mayor potencia de frecuencia entre 20 y 150 Hz [18].

Las características que se utilizarán en los algoritmos de reducciones de dimensiones están en función del tiempo, las cuales son proporcionadas por el *Grupo de Investigación en Sistemas Inteligentes de la Universidad Técnica del Norte (GISI-UTN)*.

1.8.2.1. Análisis en el dominio del tiempo

Estas características son calculadas, basadas en la amplitud de las series de tiempo de la señal electromiográficas, siendo proporcional a nivel de contracción muscular ejercido tras un movimiento [19].

A continuación, se describen algunas características de las señales electromiográficas en el dominio del tiempo expresadas por [19], [20], Se asume que se dispone de N datos correspondientes a muestras de la señal EMG.

Donde x_i representa la señal mioeléctrica en un segmento i y N denota la longitud de la señal.

- **Integral de la señal EMG (IEMG)**

Comúnmente utilizado como índice de detección del inicio de la contracción muscular en el reconocimiento de patrones bioeléctricos, Se define como la suma de los valores absolutos de la amplitud de la señal, expresada como [21].

$$IEMG = \sum_{i=1}^N |x_i| \quad (1)$$

- **Valor absoluto medio (MAV)**

Es un método de permite detectar y medir los niveles de contracción muscular Es una de las técnicas más populares utilizadas en el análisis de bioseñales, especialmente en la detección de las señales mioeléctricas para el control de prótesis y exoesqueletos, se define como el promedio del valor absoluto de la señal en un segmento i y de longitud N [21].

$$MAV = \frac{1}{N} \sum_{i=1}^N |x_i| \quad (2)$$

- **Raíz Media Cuadrática (RMS)**

Al igual que el valor medio absoluto, es otra de las técnicas más utilizadas debido a que permite caracterizar la señal de acuerdo con el estimado de energía, relacionada con la amplitud en ciertos intervalos de tiempo [21]. Su expresión viene dada por:

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (3)$$

- **Varianza de la señal EMG (VAR)**

La varianza de la señal electromiográficas es un índice de potencia [21]. Se define como un promedio del valor cuadrático de la desviación estándar de la variable, sin embargo, el valor medio de la señal mioeléctricas es cercana a 0 ($\sim 10^{-10}$), por consiguiente, la varianza de la señal se define como:

$$VAR = \frac{1}{N-1} \sum_{i=1}^N x_i^2 \quad (4)$$

- **Longitud de onda (WL)**

La longitud de la señal es una medida de la complejidad de la señal mioeléctrica. Es una variación acumulativa de la señal que puede indicar su grado de transición, es decir que es el acumulado de la distancia entre dos muestras consecutivas de la señal. Su expresión viene definida por [21]:

$$WL = \sum_{i=1}^N |x_{i+1} - x_i| \quad (5)$$

- **Cambios en el signo de la señal (SSC)**

Es una característica que representa la información en frecuencia de la señal mioeléctrica, su cálculo cuantifica el número de veces que la pendiente de la señal cambia de signo, el número de cambios entre las pendientes positivo y negativo, entre tres segmentos secuenciales es realizado con un nivel de umbral para evitar el ruido inducido [21], su expresión matemática viene dado:

$$SSC = \sum_{i=1}^N [f[(x_i - x_{i-1}) * (x_i - x_{i+1})]] \quad (6)$$

$$f(x) = \begin{cases} 1, & \text{si } x \geq \text{nivel de umbral} \\ 0, & \text{en caso contrario} \end{cases}$$

- **Amplitud de Willison (WAMP)**

La amplitud de Willison calcula el número de veces en que la diferencia resultante en amplitud de la señal mioeléctrica entre dos segmentos consecutivos, exceden un

umbral predefinido. El parámetro es un indicador de la activación de los potenciales de acción de las unidades motoras y por consiguiente de los niveles de contracción muscular [21]. El cálculo es formulado por:

$$WAMP = \sum_{i=1}^{N-1} f(|x_n - x_{n+1}|) \quad (7)$$

$$f(x) = \begin{cases} 1, & \text{si } x \geq \text{nivel de umbral} \\ 0, & \text{en caso contrario} \end{cases}$$

▪ Simple cuadrado integral (SSI)

Usa la energía de la señal EMG como una característica [21], el cálculo es formulado por:

$$SSI_k = \sum_{i=1}^N (|x_i|^2) \quad (8)$$

1.8. Reducción de dimensiones

El inconveniente que se presenta en la técnica de extracción de características mediante la aplicación de transformaciones de la señal es la alta dimensionalidad del espacio resultante, lo que implicaría conformar vectores de descripción muy grandes. Ante esto, las técnicas de reducción de dimensionalidad permiten preservar la información más relevante al tiempo que reduce la dimensión del espacio transformado [16]; procedimiento esencial para incrementar el desempeño de la clasificación. La reducción de dimensionalidad se puede lograr de dos maneras: por proyección de características o por selección de características [4].

En general se prefiere trabajar la técnica de selección de características, pues de esta manera se conserva el conjunto reducido la información contenida en cada una de las características del conjunto original. Este es un procedimiento automático que no requiere de la intervención humana para la selección de las características finales. Dentro de esta técnica el método más utilizado es el análisis de componente principal (ACP) [16], [4].

1.9.1. Análisis de componente principal (ACP)

El análisis de componentes principales (ACP) es probablemente la técnica más popular de la estadística multivariada y es utilizada en la mayor parte de las disciplinas científicas en las que manejan información cuantitativa o cualificada. Esta se encarga de reducir la dimensión de un conjunto de datos mediante el cálculo de un grupo mucho menor de variables que mejor representan el conjunto original de datos [22].

El uso de este método se ha extendido de forma veloz en los últimos años debido principalmente al aumento del volumen de información que es necesario procesar y a la distintiva capacidad de esta herramienta [23].

El método de componentes principales tiene por objeto transformar un conjunto de variables, a las que se denominan originales, en un nuevo conjunto de variables denominadas componentes principales. Estas últimas se caracterizan por estar correlacionadas entre sí y, además, pueden ordenarse de acuerdo con la información que lleven incorporada [24].

Como medida de la cantidad de información incorporada en un componente se utiliza su varianza. Es decir, cuando mayor sea su varianza mayor es la cantidad de información que lleva incorporada dicho componente. Por esta razón se selecciona como primera componente aquella que tenga mayor varianza, mientras que la última componente es la de menor varianza [24].

En general, la extracción de componentes principales se efectúa sobre variables tipificadas para evitar problemas derivativos de la escala, aunque también se puede aplicar sobre variables expresadas en desviaciones respecto a la media [25].

Si p variables están tipificadas, la suma de las variables es p , ya que la varianza de una variable tipificada es por definición 1 [25].

El nuevo conjunto de variables que se obtiene por el método de componentes principales es igual en número al de las variables originales. Es importante destacar que la suma de sus variables es igual a la suma de las variables originales [25].

1.9.2. Análisis Discriminante Lineal (ADL)

El Análisis discriminante lineal bajo el criterio de Fisher, busca las muestras que sigan patrones de distribución gaussianas para un buen funcionamiento. Dentro de sus aplicaciones más usadas en la literatura, se recomienda para tareas de aprendizaje supervisado & clasificación de clases, ya que proyecta los datos a una baja dimensión en comparación de los datos originales, pero garantiza la máxima dispersión entre clases para reducir a su mínimo posible la dispersión interna de cada clase [26]. La tarea es proyectar el conjunto de datos a una baja dimensión mediante un vector w que pueda garantizar el máximo en dispersión o separación de las clases [27] [28]. Se tiene $x_1 \dots x_n$ tuplas d -dimensionales que se etiquetan en c clases. Cada una de estas clases tiene N_c tuplas. El vector w se requiere, para encontrar $y_i = w^T x_i$ estas serán las proyecciones en baja dimensión de cada una de las tuplas.

Para ese propósito, Fisher-LDA considera maximizar el siguiente objetivo:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} \quad (9)$$

donde S_B es la "matriz de dispersión entre clases" y S_W es la "matriz de dispersión dentro de las clases". Tenga en cuenta que debido al hecho de que las matrices de dispersión son proporcionales a las matrices de covarianza, se podría haber definido J utilizando matrices de covarianza: la constante de proporcionalidad no tendría efecto en la solución. Las definiciones de las matrices de dispersión son:

$$S_B = \sum_c N_c (\mathbf{u}_c - \mathbf{u})(\mathbf{u}_c - \mathbf{u})^T \quad (10)$$

$$S_W = \sum_c \sum_{i \in c} (\mathbf{u}_c - \mathbf{u})(\mathbf{u}_c - \mathbf{u})^T \quad (11)$$

Donde, u_c es la media para cada una de las clases, u posee la media de todos los datos, N_c la cantidad de tuplas que pertenecen a la clase c . Fisher-LDA busca proyectar el vector w que pueda maximizar el "cociente" entre la base de la matriz de dispersión inter-clase y la matriz de dispersión intraclase.

Cuando w maximice la función objetivo tiene que satisfacer la siguiente condición:

$$S_B w = \lambda S_W w \quad (12)$$

Si S_W es no singular se puede desarrollar con las operaciones de valores propios para la matriz $S_W^{-1} S_B$:

$$S_W^{-1} S_B w = \lambda w \quad (13)$$

Se sustituye el resultado para obtener el nuevo $J(w)$:

$$J(w) = \frac{w^T S_B w}{w^T S_W w} = \lambda_k \frac{w_k^T S_B w_k}{w_k^T S_W w_k} = \lambda_k \text{ con } k = 1 \dots d \quad (14)$$

Del cual se tiene un w_k vector propio k de valor propio λ_k . Esto indica que, para maximizar la función objetivo de la solución, se debe tener un vector propio asociado a mayor valor propio.

1.9.3. Análisis de Componentes Independientes (ACI)

El Análisis de Componentes Independientes (ACI) es una herramienta que busca separar fuentes estadísticamente independientes a partir de mezclas obtenidas mediante diferentes sensores [5].

Siendo el conjunto inicial de entrenamiento $X = [\Phi_1, \Phi_2, \dots, \Phi_p]$, que funciona bajo las siguientes suposiciones: las fuentes son estadísticamente independientes, solo una de las fuentes puede tener una distribución gaussiana y las señales capturadas en los sensores $\{\Phi_j = \phi_j(t), j = 1, \dots, p \wedge t = 1, \dots, n\}$ son combinaciones lineales de las fuentes $s_j(t)$, para dos fuentes y dos sensores el problema se formula así: [29].

$$\begin{bmatrix} \phi_1(t) \\ \phi_2(t) \end{bmatrix}, A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, s = \begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix} \quad (15)$$

Con el objetivo de encontrar $a_{11}, a_{12}, a_{21}, a_{22}, s_1(t), s_2(t)$ para $\phi_1(t)$ y $\phi_2(t)$ conocidas. De forma general, para un número de sensores igual a la fuente, se tiene:

$$x = \begin{bmatrix} \phi_1(t) \\ \dots \\ \phi_p(t) \end{bmatrix}, A = \begin{bmatrix} a_{11} & \dots & a_{1p} \\ \dots & \dots & \dots \\ a_{p1} & \dots & a_{pp} \end{bmatrix}, s = \begin{bmatrix} s_1(t) \\ \dots \\ s_p(t) \end{bmatrix} \quad (16)$$

De tal manera, que las combinaciones lineales de las fuentes se pueden reescribir como $x = A * s$. Cuando A es encontrada, s puede ser calculada como $s = A^{-1}x$. La solución encontrada cuenta con la limitación de que tanto los signos como el orden de las fuentes no pueden ser determinados [30] [31].

Sea $y_j(t)$, la estimación de un componente independiente ($\approx s_j(t)$) y w_j una fila de la matriz estimada de transformación inversa $W \approx A^{-1}$. Un componente independiente se estima de acuerdo con la expresión $y_j(t) = w_j^T x$.

Bajo ciertas condiciones, la distribución que resulta de la suma de variables aleatorias estadísticamente independientes, es mucho más cercana a la gaussiana que cualquiera de las distribuciones de las variables originales[32]. El uso de este teorema resulta básico en la implementación del algoritmo ACI. Como $\phi_j(t)$ es una suma ponderada de los componentes en s y los componentes son no gaussianos e independientes, el problema se reduce a encontrar w_j de tal manera que la no gaussividad de $y_j(t)$ sea maximizada [33]. Una manera de medir la gaussividad, es por medio de la siguiente aproximación de negentropía [34].

$$J(y) \propto (E\{G(y)\} - E\{G(v)\})^2 \quad (17)$$

Donde \propto denota proporcionalidad; v es una variable gaussiana con media de cero y varianza unitaria, de manera que el termino $E\{G(v)\}$ es una constante; $G(y)$ es una función no cuadrática, cuya elección depende del problema, y las comúnmente usadas son: $G_1(y) = \frac{y^4}{4}$, $G_2(y) = \log(\cosh(y))$ y $G_3(y) = -\exp\left(-\frac{y^2}{2}\right)$. Recalcando que el principal interés es calcular w_j que maximice $J(w_j^T x)$.

1.9.4. Método de Regresión Lineal Múltiple (MRLM)

El análisis de Regresión Lineal Múltiple es aplicado en las siguientes posibilidades: como primera posibilidad una variable dependiente continua con un conjunto de variables categóricas y como última, una variable dependiente nominal con un conjunto de variables continua. Obteniendo como resultado, que tanto las variables dependientes como las independientes sean variables continuas medidas en intervalos [35].

Siendo $X \in \mathbb{R}^{n \times d}$, denotando la matriz de diseño donde n y d son el número de observaciones y variables de entrada respectivamente, e $Y \in \mathbb{R}^{n \times p}$, Denotando la matriz objetivo, donde p es la dimensionalidad de salida [35].

C_{xx} es la matriz de covarianza de entrada, y C_{xy} es la matriz de covarianza de entrada y salida. Por definición, $C_{xy} = C_{yx}$. Se puede probar que el problema de MLR se puede expresar de la siguiente manera: [35].

$$Av = \lambda Bv \quad (18)$$

Donde:

$$A = \begin{pmatrix} \mathbf{0} & C_{xy} \\ C_{xy} & \mathbf{0} \end{pmatrix}, B = \begin{pmatrix} C_{xx} & \mathbf{0} \\ \mathbf{0} & I \end{pmatrix} \quad (19)$$

A y B son ambas matrices simétricas, como una matriz diagonal de bloque cuyos elementos diagonales son todas matrices positivas definidas, por lo que se garantiza que todos los valores propios son reales. A tiene una forma especial, es simétrica hueca, es decir, una matriz simétrica con elementos diagonales cero, por lo tanto, los valores propios de 1 vienen en pares positivos y negativos. Primero se resuelve el problema del valor propio general descrito utilizando la descomposición de Cholesky¹ de C_{xx} y se ordena los valores propios en orden descendente. Luego se calcula la suma acumulativa de los valores propios positivos y se eligen los valores propios más grandes, de manera que $\sum_{i=1}^m \lambda_i \geq 0.99 \sum_{i=1}^{k/2} \lambda_i$, donde λ_i son los valores propios positivos de 1, y $k = d + p$. Los vectores propios correspondientes w_1, \dots, w_m son seleccionados y ordenados en una matriz $W = (w_1, \dots, w_m) \in \mathbb{R}^{(d+p) \times m}$. La entrada de la transformada de bajo rango se obtiene por $X' = XW_x$, donde W_x denota la matriz de $d \times m$ cuyas filas consisten en las primeras d filas de W . En consecuencia, la salida de la transformada de bajo rango viene dada por $Y' = XW_y$, donde W_y es la matriz $p \times m$ cuyas filas consisten en las últimas p filas de W [35].

El modelo de regresión lineal múltiple se basa en la siguiente ecuación [42]:

¹ La descomposición de Cholesky se usa principalmente para hallar la solución numérica de ecuaciones lineales $Ax = b$. Si A es simétrica y positiva definida, entonces se puede solucionar $Ax = b$ calculando primero la descomposición de Cholesky $A = LL^T$, luego resolviendo $Ly = b$ para y , y finalmente resolviendo $L^T x = y$ para x .

$$Y = \beta_0 + \beta_1 X + \cdots + \beta_p X_p + \varepsilon \quad (20)$$

Teniendo los datos de entrada con la siguiente estructura

$$\begin{array}{cccc} y_1 & x_{11} & \cdots & x_{1p} \\ y_2 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ y_n & x_{n1} & \cdots & x_{np} \end{array} \quad (21)$$

De la expresión matemática del modelo de regresión lineal general se deduce

$$y_i = \beta_0 + \beta_1 X + \cdots + \beta_p X_p + \varepsilon_i, i = 1, 2, \dots, n \quad (22)$$

Se asume que los errores $\varepsilon_i, \varepsilon_2, \dots, \varepsilon_n$ tienen distribución normal cero y varianza σ^2 , y que son independientes.

Las variables explicativas son linealmente independientes entre sí, por lo cual se plantea el modelo en forma matricial de la siguiente manera:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1p} \\ 1 & x_{21} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{np} \end{pmatrix} \cdot \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{pmatrix} \quad (23)$$

Obteniendo como resultado el modelo de regresión lineal múltiple.

$$y = \beta_0 + \beta_1 X + \cdots + \beta_p X_p \quad (24)$$

Capítulo 2

2. Marco Metodológico.

En el presente capítulo se hará una descripción de la metodología a utilizar para la adquisición de los datos, procesamiento y reducción de dimensiones. Además de los procedimientos utilizados para la validación de los métodos de reducción.

2.1. Selección de métodos de reducción de dimensiones

Se mostrará a continuación los beneficios y desventajas de cada método, de acuerdo con estas características se seleccionarán dos.

ACP es un método apropiado en tareas de compresión de datos debido a que realiza una combinación lineal, proyectando los datos sin perder todas sus propiedades, para describirlos en un espacio de una dimensión más baja, sin embargo, tiene deficiencia al intentar reducir un número muy grande de dimensiones.

ADL tiene un óptimo rendimiento en tareas de clasificación gracias al punto de corte del modelo discriminante que proyecta una robusta regla de decisión y es efectivo en separabilidad intraclases. Se usa para reducir dimensiones, pero su fortaleza radica en proyectar un buen desempeño en tareas de clasificación conservando la separabilidad entre clases.

ICA representa los datos multivariantes en componentes estadísticamente independientes. Sin embargo, no se logra reducir en mucha proporción la dimensión del espacio inicial de características, aunque la tasa de acierto se incrementa considerablemente. A pesar de esta limitante es recomendable usar este método para variables con distribución gaussiana

RLM reduce considerablemente, llegando a ser más pequeña la dimensión que la estimada por ACP. Siendo la reducción del 86% con el método RLM y del 71% del método ACP.

A partir de los métodos referenciados anteriormente se seleccionaron dos métodos para realizar su respectiva programación en el software *MatLab*. El primer método es el *Análisis de Componente Principal (ACP)* y como segundo el *Método De Regresión Lineal Múltiple (MRLM)*. Se comparará la eficiencia en la reducción de dimensiones a partir de los procesos de validación. Vale recalcar que el primer método requiere poco recurso y consta de una programación relativamente fácil a comparación del último método que requiere de más recursos y es más compleja la elaboración del algoritmo.

2.2. Ingreso de datos

El primer paso que se realiza es el ingreso de datos, cargando las cuatro matrices en Matlab, siendo cada una de un movimiento (Extensión, Flexión, Pronación y Supinación) del brazo. Cada matriz está conformada por quinientas filas y cuarenta y cuatro columnas, es decir que contiene quinientas muestras de cuatro músculos, y en cada toma del musculo se obtiene 11 características como se muestra en la Tabla 2.1.

Tabla 2.1 Matriz de características de un movimiento.

Muestra/Característica	Característica 1	Característica 2	...	Característica 11
Muestra 1	M1C1	M1C2	...	M1C3
⋮	⋮	⋮	⋮	⋮
Muestra 500	M500C1	M500C2	...	M500C2

2.3. Modificación de la Base de Datos

2.2.1. Normalización

Se obtienen los valores máximos de cada columna con el comando “*max*”, posterior se consiguen los valores de cuantas filas y columnas tiene la matriz a normalizar con el comando “*size*”.

Se realizan dos ciclos “*for*” anidados, siendo el primero para las filas y el segundo para las columnas; dentro de los ciclos se le asigna a una matriz auxiliar llamada “*Xn*” los valores de la matriz sin normalizar en la posición de los ciclos, divididos para el valor máximo de la columna.

2.2.2. Matriz de entrenamiento

Para crear la matriz de entrenamiento se realizan dos ciclos “*for*” anidados, siendo el primer ciclo el que recorre las filas y el segundo las columnas. Se colocan las siguientes limitaciones: para el primer ciclo se comienza en uno y termina en doscientos cincuenta, y el segundo ciclo comienza en uno y termina en once.

Dentro del ciclo anidado se le asigna a una nueva matriz auxiliar llamada “*X*” todos los valores de las doscientas cincuenta muestras con las once características de cada movimiento, obteniendo como resultado una matriz de mil muestras con once características.

2.2.3. Matriz de prueba

Para la elaboración de la matriz de prueba se usa un ciclo “*for*” para las columnas con una limitante de doscientos cincuenta y otro ciclo “*for*” para las filas teniendo como limitante de once. Generando una matriz de mil muestras con once características.

En la Tabla 2.2. se muestra la distribución de la matriz de entrenamiento, para la matriz de prueba se utilizaron desde la muestra 251 hasta la 500 de cada movimiento.

Tabla 2.2 Matriz de características normalizada

Muestra/Característica	Característica 1	Característica 2	...	Característica 11
Extensión (250 muestras)	Valor (0-1)	Valor (0-1)	Valor (0-1)	Valor (0-1)
Flexión (250 muestras)	:	:	:	:
Pronación (250 muestras)	Valor (0-1)	Valor (0-1)	Valor (0-1)	Valor (0-1)
Supinación (250 muestras)	Valor (0-1)	Valor (0-1)	Valor (0-1)	Valor (0-1)

Las funciones de Normalización, Matriz Entrenamiento y Matriz Prueba se realizaron a partir de los flujogramas de la Figura 2.1, además se adjunta las líneas de código en el Anexo Algoritmos de Reducción de Dimensiones con el Título “Normalización”, “Creación de Base de Datos-Base de datos sin procesamiento”.

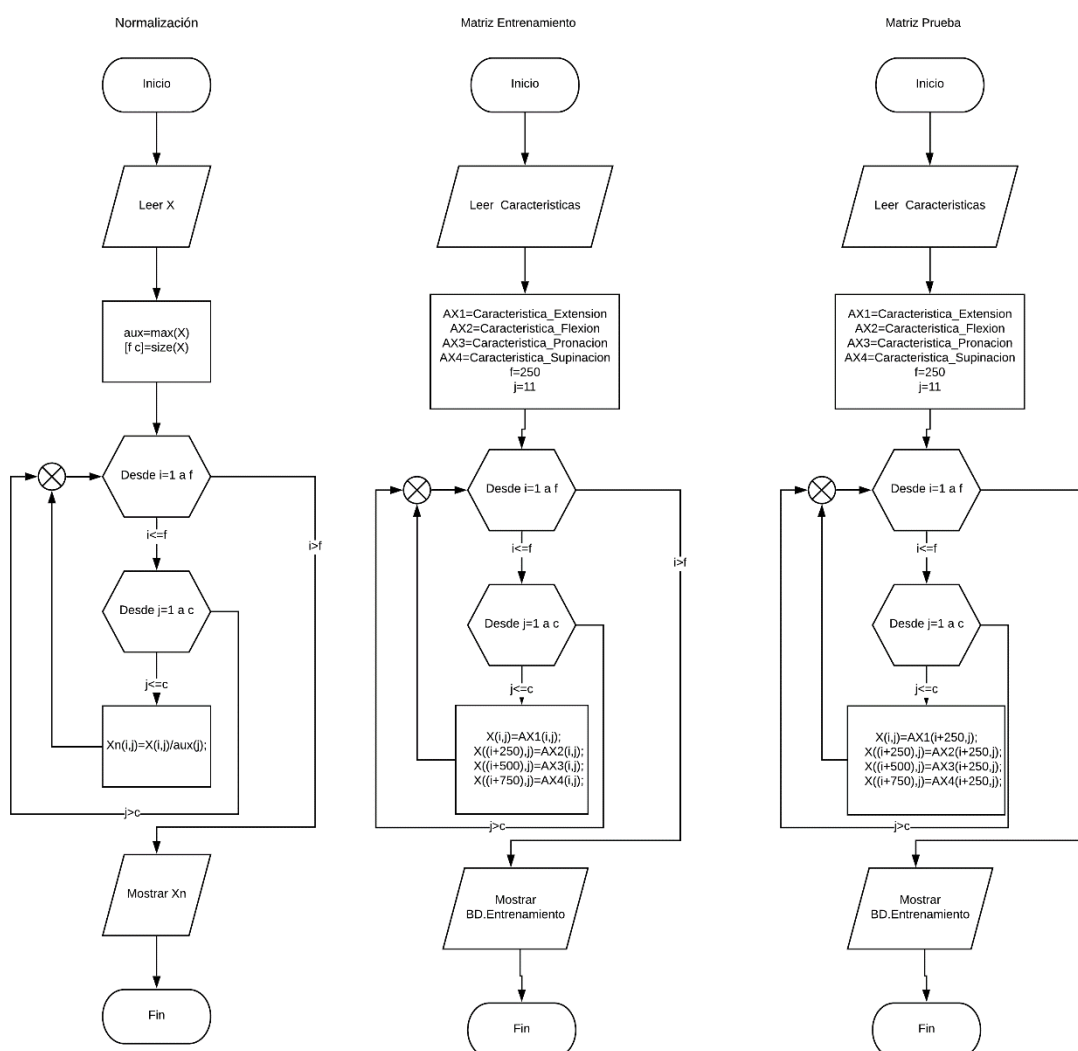


Figura 2.1 Diagramas de flujo de Funciones de Normalización, Matriz Entrenamiento y Matriz Prueba

2.4. Métodos de Reducción de Dimensiones

Se presenta la programación de los dos algoritmos de los métodos de reducción de dimensiones.

2.3.1. Análisis de Componente Principal

Se realiza una condición, en el caso de que no se especifiquen el número de dimensiones, se asigna el número dos. Posterior se obtiene el valor medio de características ingresadas para restarlo a los valores de las características originales. Luego se obtiene la matriz de covarianza para utilizar estos valores en el comando “*eig*” el cual da como resultado los valores y los vectores propios, los cuales se ordenan en orden descendente. Se ejecuta el mapeo de datos, llevándose a cabo a través de la multiplicación de la resta de los valores medios y las características originales por los vectores propios de la matriz de covarianza. Obteniendo “*mu*” utilizamos el comando “*pca*” el cual es la varianza total de cada característica, posteriormente dicho resultado entra a un conjunto de ciclos anidados y condicionales, en el cual se identifica qué característica tiene mayor varianza y se le asigna el número de la característica en un vector nuevo; teniendo como salida un vector con los números de las características ordenadas descendentemente con respecto a su varianza como se puede observar en la Figura 2.2. y en el Anexo Algoritmos de Reducción de Dimensiones con el Título “Análisis de Componente Principal”.

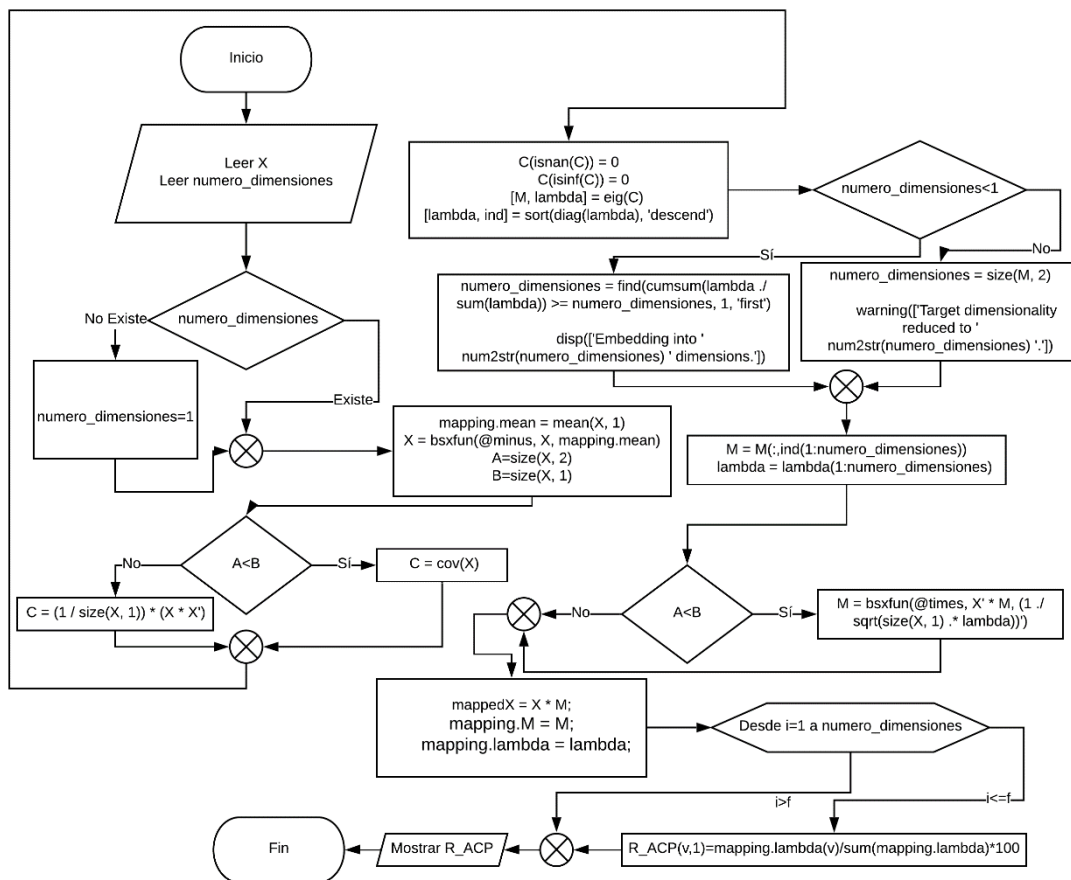


Figura 2.2 Diagrama de flujo del Análisis de Componente Independiente.

2.3.2. Método de regresión lineal múltiple

Se utiliza el comando de Matlab “*stepwiselm*” el cual es un método paso a paso para realizar una regresión lineal múltiple, agregando y eliminando predictores. Teniendo como resultado un modelo de regresión lineal, del cual obtenemos un arreglo en el que podremos observar los coeficientes, seguidamente se realizan dos ciclos “*for*” con condicionales “*if*” con el objetivo de obtener los coeficientes y las posiciones de los mismos, luego entra a un condicional “*switch*” para identificar qué característica tiene mayor incidencia en el modelo y obtendremos como salida las tres características con mayor influencia en el modelo como se muestra en la Figura 2.3 y Figura 2.4. y en el Anexo Algoritmos de Reducción de Dimensiones con el Título “Método de Regresión Lineal Múltiple”.

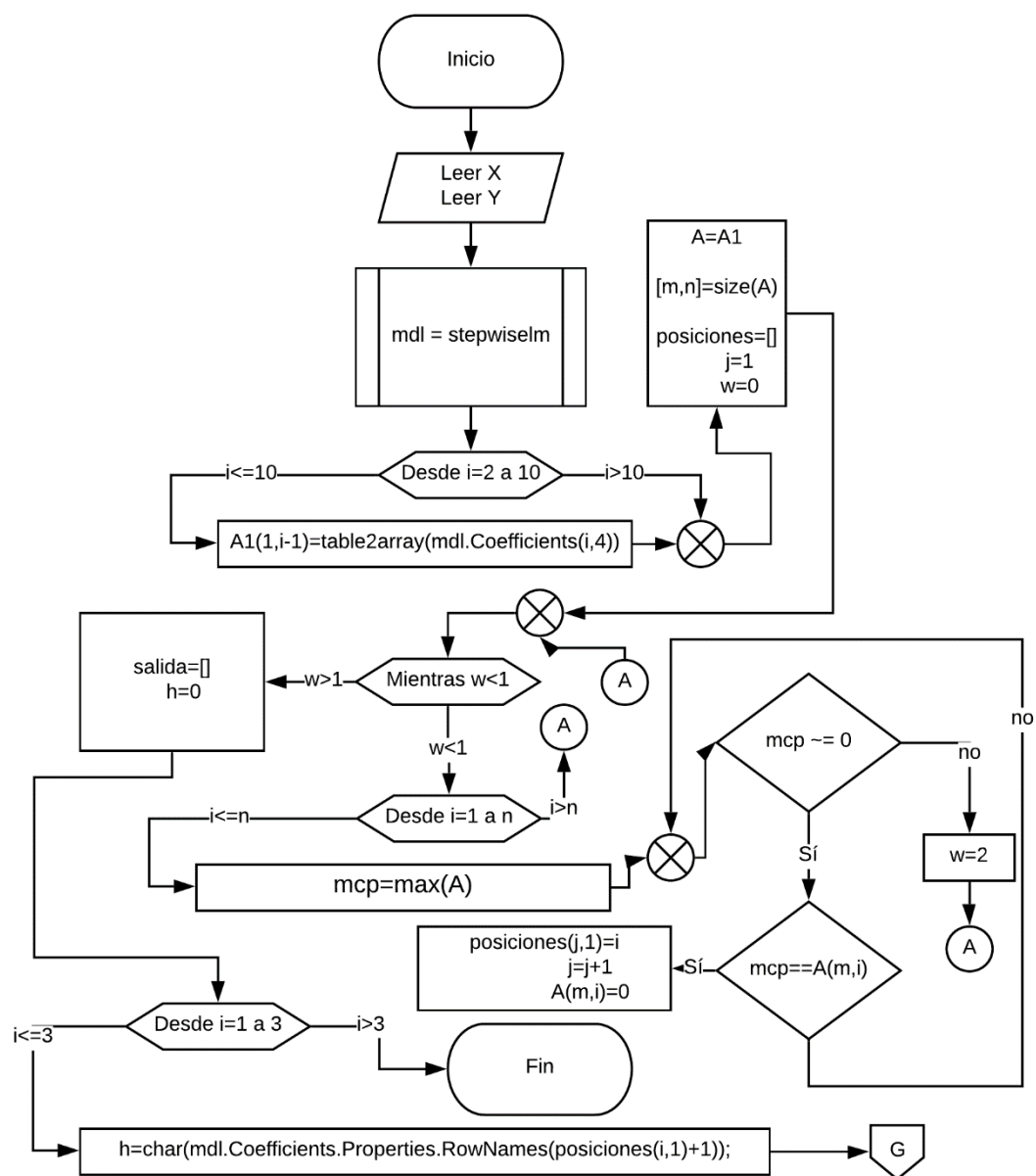


Figura 2.3 Pagina 1 Diagrama de Flujo de Método de Regresión Lineal Múltiple

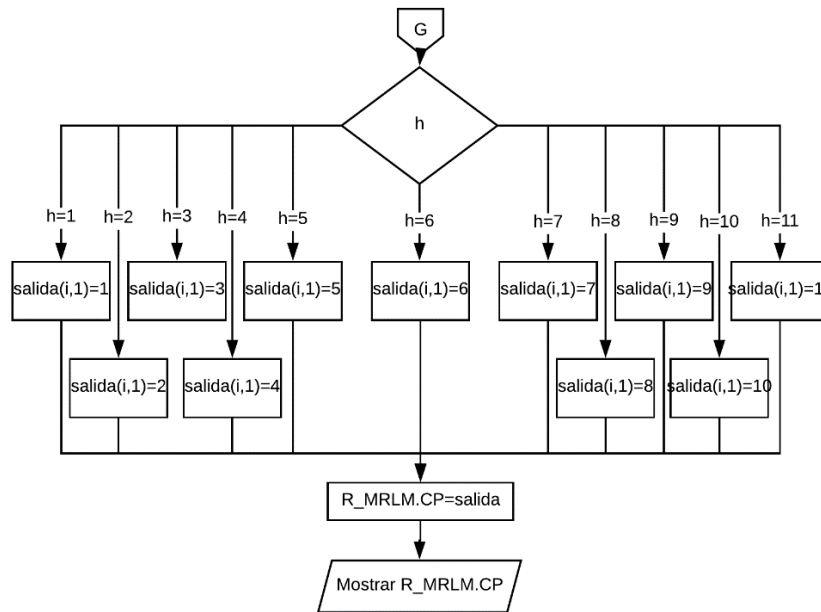


Figura 2.4 Pagina 2 Diagrama de Flujo de Método de Regresión Lineal Múltiple

2.5. Obtención bases de datos con dimensión reducida

Se crea dos bases de datos, la primera que contendrá las características reducidas por el método de Análisis de Componente Independiente y la segunda por el método de Regresión Lineal Múltiple como se muestra en la Figura 2.5. y en el Anexo Algoritmos de Reducción de Dimensiones con el Título “Creación de Base de Datos-Base de datos procesada”.

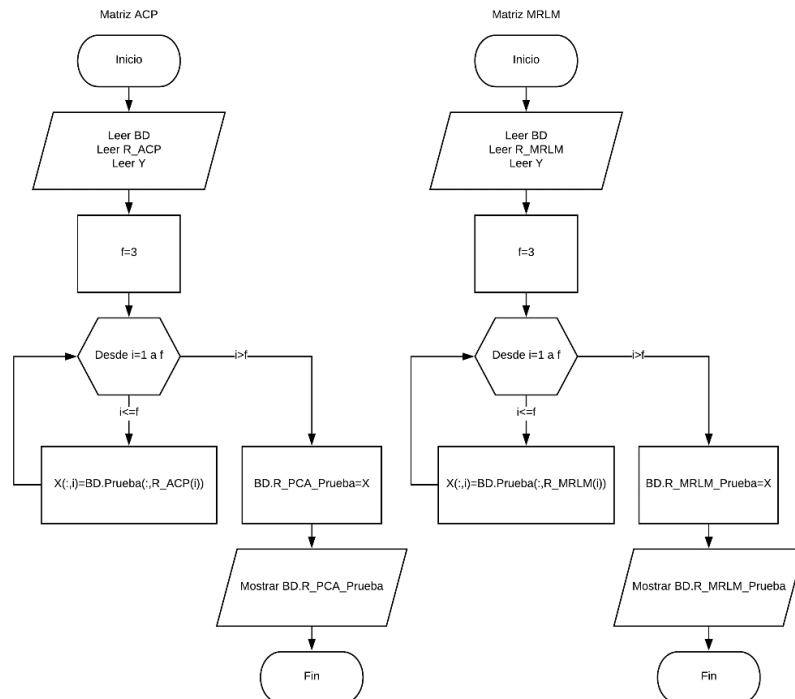


Figura 2.5 Diagrama de Flujo de las Bases de Datos de dimensión reducida

2.6. Métodos de clasificación

Para la clasificación se utilizó el “*PR toolbox*” de Matlab. Como primer paso se procede a cargar las bases de datos, se parametriza la información con la función “*prdataset*” y posterior se declara la variable que contendrá el número de iteraciones que realizarán los métodos de clasificación. Posteriormente se llama a las funciones propias de la herramienta utilizada, siendo “*ldc*”, “*qdc*”, “*knnc*” los métodos Clasificador lineal basado en densidades normales, Clasificador cuadrático basado en densidades normales y Clasificador K Bayesiano más cercano respectivamente (Figura 2.6)(Anexo Algoritmos de Clasificación). Dando como resultado el error de clasificación de cada método, siendo estos valores los determinantes para el análisis del siguiente capítulo.

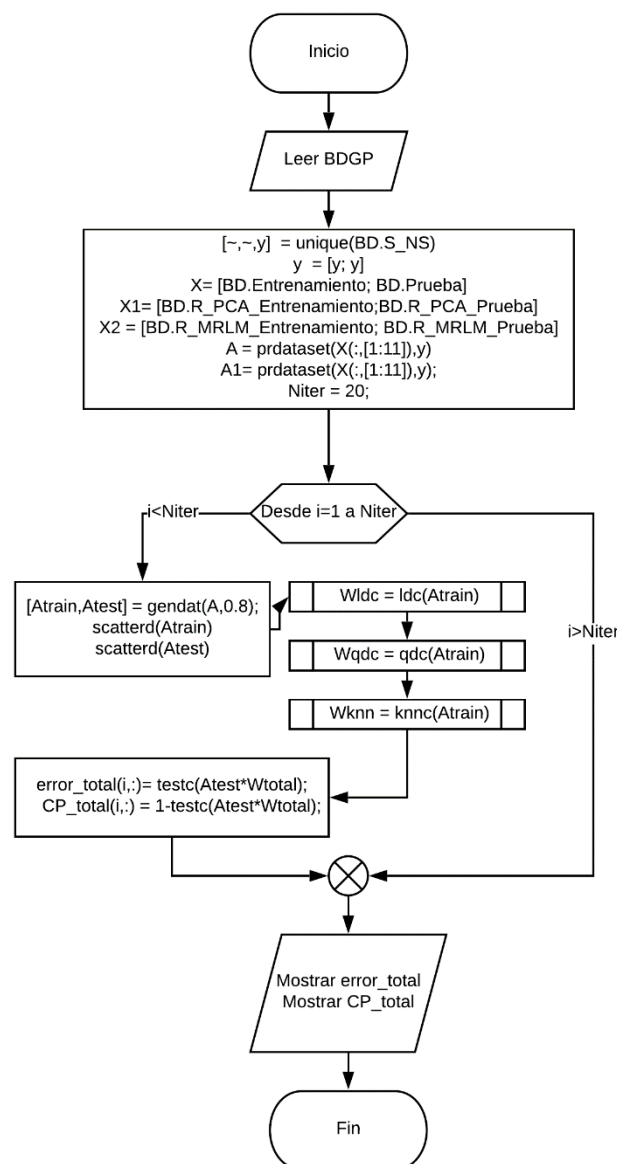


Figura 2.6 Diagrama de Flujo de Clasificación de Características

En la Figura 2.7 se puede observar los procesos a los cuales se somete las características. Comenzando con la lectura de la base de datos, posterior se normaliza y se crea bases de datos para el entrenamiento y las pruebas. Estas matrices son utilizadas para la reducción de dimensiones y la clasificación. De esta manera se obtiene como resultado las tres características más importantes, los porcentajes de acierto y error de los diferentes métodos de clasificación y el tiempo de ejecución.

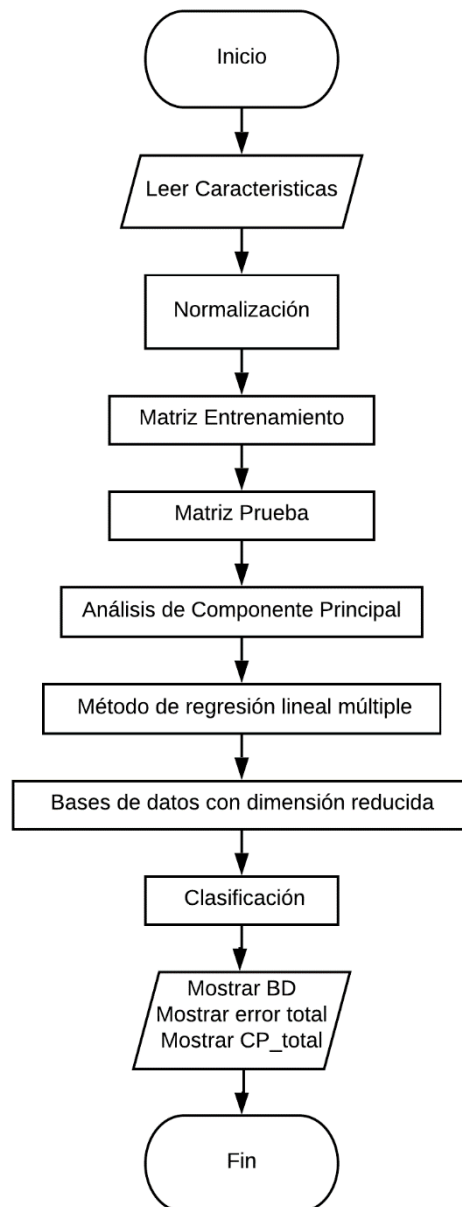


Figura 2.7 Diagrama de flujo general

Capítulo 4

3. Análisis y Resultado

Una vez aplicados los métodos de reducciones, obtenemos como resultado tres características importantes como se muestra en la Tabla 3.1.

Tabla 3.1 Métodos de reducciones con sus tres principales características

Método	Nombre de característica
ACP	1) WAMP
	2) RMS
	3) MAV
MRLM	1) WAMP
	2) EDT
	3) SSI

Posterior a reducir dimensiones se aplicarán dos métodos de validación.

Como primer método de validación de los algoritmos de reducción de dimensiones se utilizó la herramienta “*Pr toolbox*” de *MatLab*, aplicando tres métodos de clasificación con veinte iteraciones cada uno.

- Clasificador lineal basado en densidades normales
- Clasificador cuadrático basado en densidades normales
- Clasificador K bayesiano más cercano

Se aplicaron los tres métodos de clasificación a las once características iniciales, consiguiendo los porcentajes de acierto y error que se muestran en la Figura 3.1, en la cual se observar que a pesar de que se usa las once características y mil muestras, no se obtiene un porcentaje de acierto del cien por ciento en el método de clasificación uno y tres, siendo el procedimiento tres con mayor porcentaje de error (2.57%).

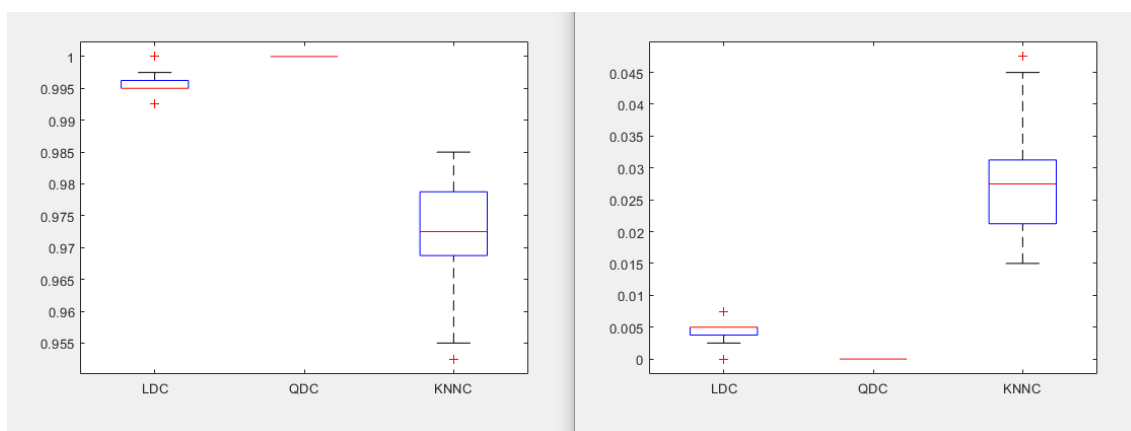


Figura 3.1 Acierto y error de la clasificación de la base de datos de once características.

Para validar los dos métodos de reducción de dimensiones los resultados no tienen que alejarse a los valores obtenidos en la clasificación de las once características.

Se realiza la primera validación al método de análisis de componente principal como se muestra en la Figura 3.2, en la cual se aprecia que tiene como porcentaje mínimo de clasificación de 93% (LDC) y máximo de 99% (QDC).

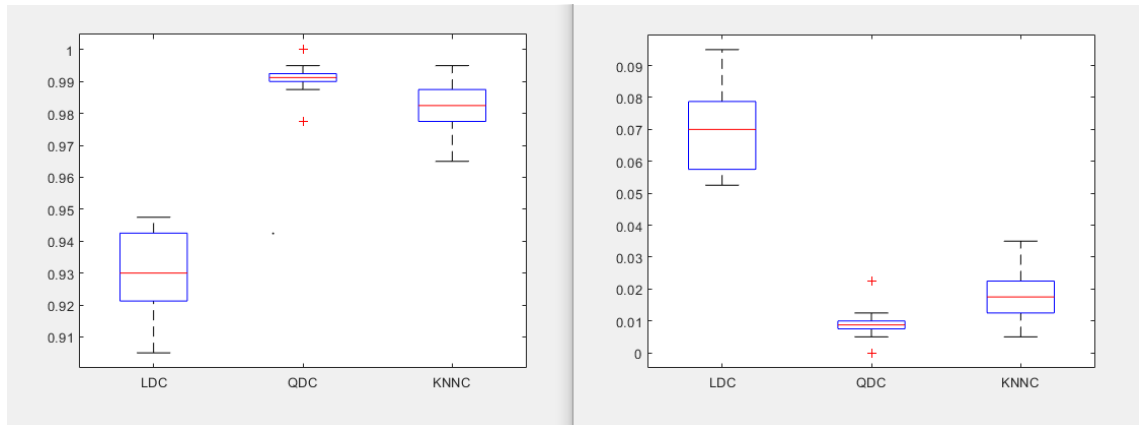


Figura 3.2 Acierto y error de la clasificación de la base de datos de dimensión reducida de tres características por método de *ACP*.

Para el método de regresión lineal múltiple se obtuvo como resultado (

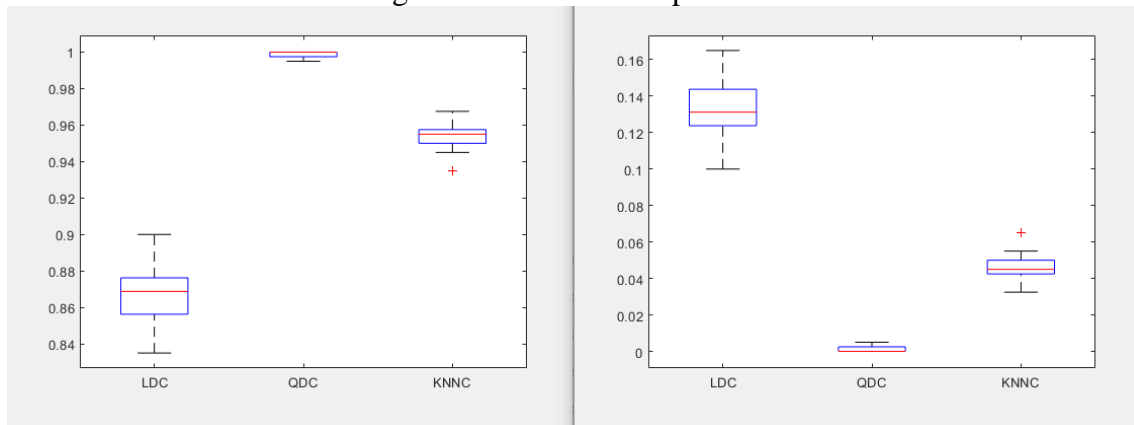


Figura 3.3) que el porcentaje mínimo de clasificación es de 86% (LDC) y el máximo es de 100% (QDC).

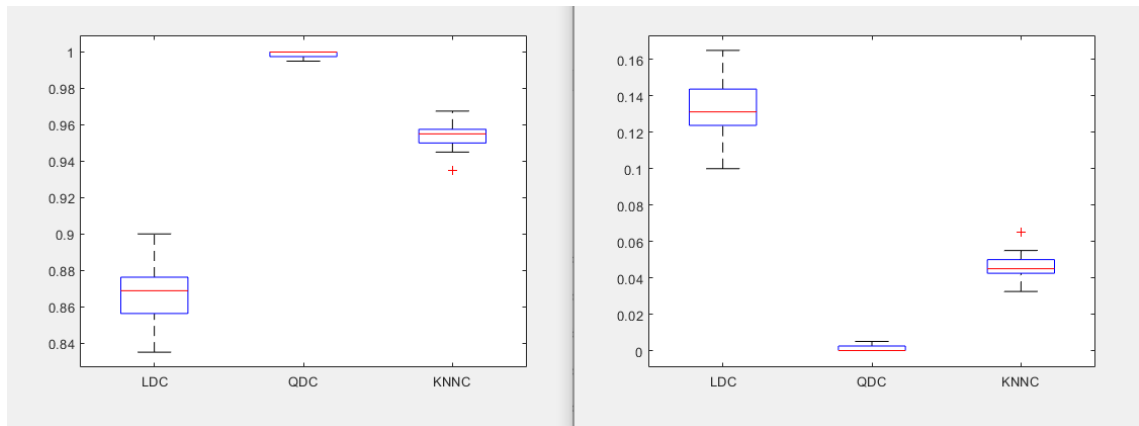


Figura 3.3 Acierto y error de la clasificación de la base de datos de dimensión reducida de tres características por *MRLM*.

En la Figura 3.4 se muestra la comparación de los porcentajes de acierto de clasificación de las once características y de los dos métodos de reducción de dimensiones, en el cual se aprecia que el comportamiento de la clasificación de los métodos de reducción de dimensiones se asemeja a la clasificación de las once características. Teniendo una apropiada clasificación con tres características de un total de once.

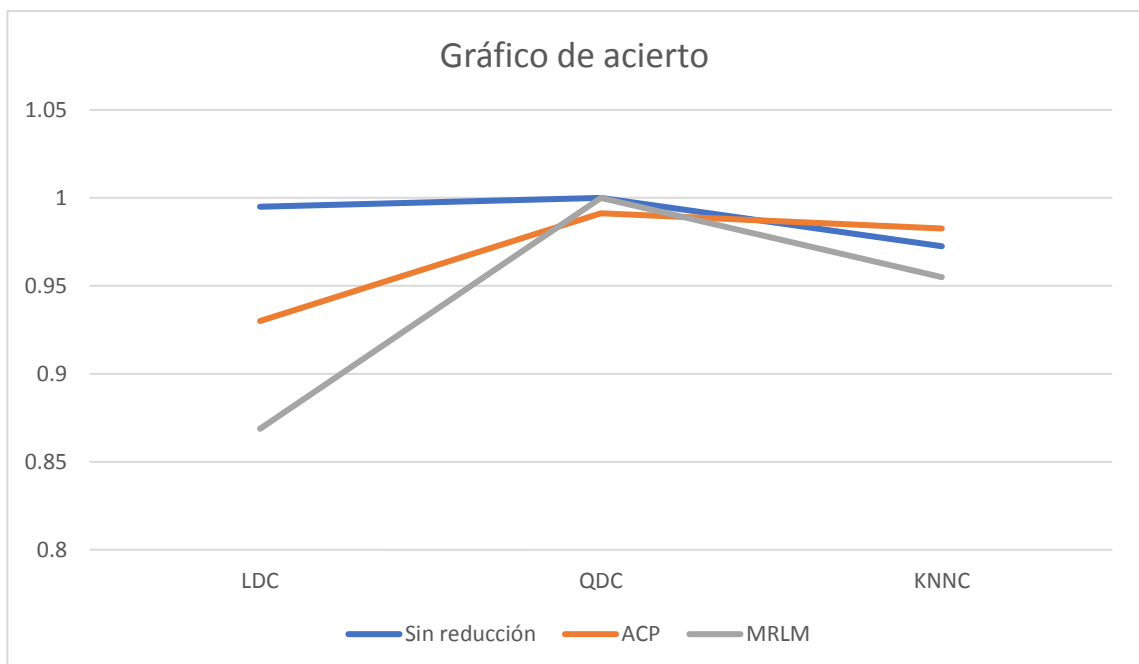


Figura 3.4 Comparativa de porcentaje de acierto de clasificación de base de datos de once dimensiones, dimensión reducida por ACP y dimensión reducida de *MRLM*.

Como segundo parámetro de validación es el tiempo de ejecución de los algoritmos de reducción de dimensiones y de los métodos de clasificación como se muestra en la Tabla 3.2. Se realiza la comparación del tiempo de ejecución de tres bases de datos, siendo la primer base de datos sin reducción de dimensiones (once características), la segunda base de datos compuesta de tres características por el método de reducción de Análisis de

Componente Principal y, la tercer base de datos conformada de tres características por el Método de Regresión Lineal Múltiple.

Tabla 3.2 Tiempo de procesamiento

Base de datos	Tiempo de reducción de dimensión (ms)	Tiempo de clasificación (ms)	Total (ms)
Once características	0	49.26	49.26
Método Análisis de Componente Principal	0.10	19.97	20.07
Método de Regresión Lineal Múltiple	6.31	20.75	27.06

Para medir el tiempo de procesamiento se utilizó la función “*tic*” (registra el tiempo interno en la ejecución) al inicio de cada método y “*toc*” (muestra el tiempo transcurrido) al final de este.

El tiempo total de la primer base de datos es 49.26ms, dicho tiempo solo fue del proceso de clasificación ya que no se redujo dimensiones. En la segunda base de datos se demoró 0.10ms en reducir las dimensiones y 19.97ms en clasificarla, teniendo como tiempo total de 20.07ms. Para la última base de datos se obtuvo uno tiempo de reducción de 6.31ms y clasificación de 20.75ms, dando como tiempo total de 27.06ms.

Para la programación de los algoritmos de reducciones de dimensiones y los métodos de clasificación se utilizó el software “*MatLab 2018b*”. Como referencia del tiempo de procesamiento de estos métodos se utilizó un computador con las características que se muestran en la Tabla 3.3.

Tabla 3.3 Característica del computador

	Procesador	Frecuencia	Memoria RAM
Computador	i7 6700K	4.00 GHz	16385 Mbyte

Conclusión y Recomendaciones

Se seleccionó el método de *ACP* ya que los requisitos son mínimos de hardware, corto tiempo de ejecución y es el método más utilizado en la literatura. Y como segundo método el *MRLM*, siendo lo opuesto al primero, se requiere mayor capacidad de hardware, tiempo de ejecución. Comprobando si es justificable sacrificar hardware y tiempo para tener mejor resultado.

A partir de los métodos de reducción de dimensiones implementados se comprobó que funcionaron adecuadamente, lográndose una reducción de 11 a 3 características como principales al momento de identificar el movimiento de la muñeca.

Los porcentajes de aciertos de las clasificaciones a partir el método de reducción de análisis de componente principal (*ACP*) son desde el 93% a 99% de acierto y del método de regresión lineal múltiple (*MRLM*) son desde el 87% hasta 100% de acierto.

Con la implementación de los métodos de reducción de dimensiones el tiempo de ejecución de todos los algoritmos, se pudo reducir de 49,29 ms a 20,06 ms (40.69%); obteniendo de esta manera resultados más rápidos, lo cual posibilita la futura implementación a nivel de microprocesador y la ejecución de otras tareas en los tiempos liberados.

Para la reducción de dimensiones en señales mioelectricas es mejor el método de análisis de componente principal en comparación del método de regresión lineal múltiple, ya que se ejecuta un 98.33% más rápido con un acierto mínimo de clasificación del 93%. Sin embargo, si se realiza una clasificación cuadrática se recomienda utilizar el método de regresión lineal múltiple ya que tiene un acierto del 100% a pesar de que se sacrifica el tiempo de ejecución.

Referencia Bibliografica

- [1] K. A. Farry, I. D. Walker, and R. G. Baraniuk, "Myoelectric teleoperation of a complex robotic hand," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 775–788, 1996.
- [2] E. G. S. Álvaro Ángel Orozco Gutiérrez, Gustavo Andrés Betancourt Orozco, "Determinación de movimientos a partir de señales electromiográficas utilizando máquinas de soporte vectorial .," *Rev. Médica Risaralda*, vol. 11, no. c, p. 15, 2005.
- [3] R. Morales pinzón, D. Á. Morales, and V. H. Grisales, "Caracterizacion de Señales Electromiográficas Para la discriminación de Seis Movimientos de la Mano," *Sci. Tech.*, no. 42, pp. 278–283, 2009.
- [4] H. A. Romo, J. C. Realpe, and P. E. Jojoa, "Surface EMG Signals Analysis and Its Applications in Hand Prosthesis Control," *Av. en Sist. e Informática*, vol. 4, no. 1, pp. 127–136, 2007.
- [5] P. Issn, "Reducción de dimensiones para clasificación de datos multidimensionales usando medidas de información," no. 32, pp. 181–186, 2006.
- [6] J. L. Correa-Figueroa, E. Morales-Sánchez, J. A. Huerta-Ruelas, J. J. González-Barbosa, and C. R. Cárdenas-Pérez, "Sistema de adquisición de señales SEMG para la detección de fatiga muscular," *Rev. Mex. Ing. Biomed.*, vol. 37, no. 1, pp. 17–27, 2016.
- [7] A. G. Ganong William, "Manual de fisiología médica," 1980.
- [8] G. Arthur, "Neurociencia básica," *Editor. médica Panam.*, 1997.
- [9] A. Carlos, "El Abecé de la electroneuromiografía clinica," *Editor. ciencias médicas*, 2003.
- [10] K. Peter, "The abc of emg," 2005.
- [11] L. C. Bigland Brenda, "Motor unit activity in the voluntary contraction of human muscle," 1954.
- [12] I. de C. Leonor, "Biofísica celular," 2010.
- [13] H. Henry, "Fisiología para la rehabilitación," 1983.
- [14] L. Elena and D. E. L. Carmen, "MIEMBROS SUPERIORES E INFERIORES RECOPIACIÓN."
- [15] S. C. Weissman BN, "Orthopedic Radiology," *Philadelphia W.B. Saunders Co*, 1986.
- [16] A. S. I. Martínez A, Muñoz L, "Tema 12: Muñeca," *Evanescencia la Esc. pública*, 2014.

- [17] C. Eduardo, M. Gonzalez, M. Benet, R. Fernando, and M. Martínez, “El complejo articular de la muñeca : aspectos anatófisiológicos y biomecánicos , características , clasificación y tratamiento de la fractura distal del radio The Wrist Joint Complex : Anatomical , Physiological and Biomechanical Aspects , Characteristics,” pp. 430–446, 2016.
- [18] W. R, “La muñeca del adulto: control de las fracturas,” *Nursing (Lond).*, 2014.
- [19] V. V. A., “Lecciones básicas de biomecánica del aparato locomotor,” *BerlínSpringer Sci. Bus. Media*, 2001.
- [20] P. Trinidad Ríos S, Parra Ramírez PA, P. C. C. Núñez LF, Quintana Rodríguez I, and M. C. SL, “Anatomía de la Muñeca y Mano: ATLAS RADIOLÓGICO,” *Madrid Soc. Española Radiol. Médica*, 2012.
- [21] U. Técnica, D. E. L. Norte, F. D. E. Ingeniería, E. N. Ciencias, and I. D. E. L. A. Obra, ““ [1] U. Técnica, D. E. L. Norte, F. D. E. Ingeniería, E. N. Ciencias, and I. D. E. L. A. Obra, “DISPOSITIVO MECATRÓNICO PARA REHABILITACIÓN PASIVA DE MUÑECA .” 2017.DISPOSITIVO MECATRÓNICO PARA REHABILITACIÓN PASIVA DE MUÑECA .” 2017.
- [22] D. ZeccaM., MiceraS., CarrozaM., “Control Of Multifunctional Prosthetic Hands By Processing The Electromyographic Signal,” *Crit. Rev. Biomed. Eng.*, 2002.
- [23] H. A. Romo, J. C. Realpe, and P. E. Jojoa, “Análisis de Señales EMG Superficiales y su Aplicación en Control de Prótesis de Mano Surface EMG Signals Analysis and Its Applications in Hand Prosthesis Control,” vol. 4, no. 1, 2007.
- [24] Antonio Camacho, “Modelado y optimización con Algoritmos Genéticos de la fuerza muscular a partir de EMG de superficie para la estimación del par articular durante la flexoextensión del codo.,” pp. 1–24, 2014.
- [25] J. Brazeiro, S. Petraccia, and M. Valdés, “Mano Controlada por Señales Musculares,” p. 161, 2015.
- [26] C. A. Quinayás-burgos and C. A. Gaviria-lópez, “Sistema de identificación de intención de movimiento para el control mioeléctrico de una prótesis de mano robótica,” vol. 19, no. 1, pp. 27–50, 2015.
- [27] A. López Delis and A. F. Ruiz-olaya, “Métodos Computacionales para el Reconocimiento de Patrones Mioeléctricos en el Control de Exoesqueletos Robóticos : una Revisión,” *Inge@uan*, vol. 3, no. October 2012, pp. 42–59, 2017.
- [28] C. O. C. Chong, “Analysis of principal functional components in economic time series Análisis de componentes principales funcionales en series de tiempo económicas,” vol. 3, no. 2, 2015.
- [29] D. H. Peluffo-Ordoñez, “Short review of dimensionality reduction methods based on stochastic neighbor embedding,” *Adv. Self-Organizing Maps Learn. Vector Quantization*.
- [30] Jolliffe, “Principal Component Analysis,” *New York Springer*, vol. 2da ed, 2018.

- [31] S. de la F. Fernández, “Análisis de componentes principales,” *Univ. Auton. Madrid*, 2010.
- [32] M. Sugiyama, “Semi-supervised local Fisher discriminant analysis for dimensionality reduction,” *Mach. Learn. is an Int. forum Res. Comput. approaches to Learn. - Springer*.
- [33] P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.*, vol. 15.
- [34] H. B. Ramires, “Descubrimiento de conocimiento en bases de datos: La perspectiva de la visualización inteligente de la información.”
- [35] and E. O. R. Vigário, J. Särelä, V. Jousmäki, M. Hämmäläinen, “Independent component approach to the analysis of eeg and meg recordings,” *IEEE Trans. Biomed. Eng.*, vol. 47, n.
- [36] A. Hyvärinen and E. Oja, “A fast fixed-point algorithm for independent component analysis,” *Neural Computation*, no. 9. *Helsinki Univ. Technol. Lab. Comput. Inf. Sci.*, 1997.
- [37] and M. J. B. S. Romero, M. A. Mañanas, J. Riba, A. Morte, S. Giménez, S. Clos, “Nuevo método automático de filtrado ocular en señales eeg espontáneas basado en análisis de componentes independientes,” *XXV Jornadas Automática, Ciudad Real*.
- [38] and E. O. A. Hyvärinen, J. Karhunen, “Independent Component Analysis,” *I. John Wiley Sons, Ed. Wiley - Intersci.*, 2001.
- [39] and Y. J. Z. Wei, L. Jin, “Independent component analysis,” *Dep. Stat. Stanford Univ.*
- [40] P. Hubka, V. Rosik, J. Zdinak, M. Tysler, and I. Hulín, “Independent component analysis of electrogastrographic signals,” *n Meas. Sci. Rev.*, vol. 5, 2005.
- [41] and H. K. M. Borga, T. Landelius, “A unified approach to pca, pls, mlr and cca,” *Linköping Univ. Dep. Electr. Eng. Tech. Rep*, 1992.
- [42] S. Nakagawa, “Regresión Lineal Múltiple Introducción,” *El Model. estimación los parámetros, contrastes*, 2012.

Anexos

Algoritmos

Reducción de dimensiones

Inicialización de datos

```
clear all, close all, clc
BD=GPBD();
% ACP Gabriel Paredes
tic;
[R_ACP] = GPACP(BD.Entrenamiento, 11);
tacc=toc;
% MRLM Gabriel Paredes
Y=[ones(250,1);2*ones(250,1);3*ones(250,1);4*ones(250,1)];
tic;
[R_MRLM]=GPMRLM(BD.Entrenamiento,Y);
tmrlm=toc;
BD=GPBDP(BD,R_ACP.CP,R_MRLM.CP,Y);
```

Algoritmos

Análisis de Componente Principal

```
function [R_ACP] = GPACP(X, numero_dimensiones)

% Si no existe la variable de entrada 'numero_dimensiones' se le proporciona el
valor de 2
if ~exist('numero_dimensiones', 'var')
    numero_dimensiones = 2;
end

% Obtener el valor medio de las características ingresadas
mapping.mean = mean(X, 1);
% Se realiza la resta de los valores de las características (X) y los valores
medios
X = bsxfun(@minus, X, mapping.mean);

% Se obtiene la matriz de covarianza
if size(X, 2) < size(X, 1)
    C = cov(X);
else
    C = (1 / size(X, 1)) * (X * X');
end

% Realizar eigendecomposition de C
C(isnan(C)) = 0;
C(isinf(C)) = 0;
[M, lambda] = eig(C);

% Ordenar los eigenvectores en orden descendente
[lambda, ind] = sort(diag(lambda), 'descend');
```

```

if numero_dimensiones < 1
    numero_dimensiones = find(cumsum(lambda ./ sum(lambda)) >= numero_dimensiones,
1, 'first');
    disp(['Embedding into ' num2str(numero_dimensiones) ' dimensions.']);
end
if numero_dimensiones > size(M, 2)
    numero_dimensiones = size(M, 2);
    warning(['Target dimensionality reduced to ' num2str(numero_dimensiones) '.']);
end

M = M(:,ind(1:numero_dimensiones));
lambda = lambda(1:numero_dimensiones);

% Aplicar mapeo de datos
if ~(size(X, 2) < size(X, 1))
    % Normalizar para obtener los eigenvectores de la matriz de covarianza
    M = bsxfun(@times, X' * M, (1 ./ sqrt(size(X, 1) .* lambda)));
end
mappedX = X * M;

% Almacenamiento de variables de salida
mapping.M = M;
mapping.lambda = lambda;

% Obtención de peso de los componentes principales
for v = 1.0:1.0:numero_dimensiones
    mapping.scores(v,1)=mapping.lambda(v)/sum(mapping.lambda)*100;
end
%Score_ACP.Porcentaje=mapping.scores;

```

Método de determinación de los números de incidencia

```

[A0,A0,A0,A0,A0,A]=pca(X);
clear A0;
[m,n]=size(A);
posiciones=[];
j=1;
w=0;
while w <1
for i=1.0:1.0:n
mcp=max(A);
    if mcp ~= 0
        if mcp==A(m,i)
            posiciones(j,1)=i;
            j=j+1;
            A(m,i)=0;
        end
    else
        w=2;
    end
end
end
%Score_ACP.NumeroCP=posiciones;

for i=1:1:3
    R_ACP.CP(i,1)=posiciones(i,1);
end

```

```

R_ACP.PCP(i,1)=mapping.scores(i,1);
end
end

```

Método de Regresión Lineal Múltiple

```

function [R_MRLM]=GPMRLM(X,Y)

mdl = stepwiselm(X,Y,'pEnter',0.06);
A1=[];
for i=2:1:10
A1(1,i-1)=table2array(mdl.Coefficients(i,4));
end
A=A1;
[m,n]=size(A);
posiciones=[];
j=1;
w=0;
while w <1
for i=1.0:1.0:n
mcp=max(A);
if mcp ~= 0
if mcp==A(m,i)
posiciones(j,1)=i;
j=j+1;
A(m,i)=0;
end
else
w=2;
end
end
end
salida=[];
h=0;
for i=1:1:3
h=char(mdl.Coefficients.Properties.RowNames(posiciones(i,1)+1));
switch h
case 'x1'
salida(i,1)=1;
case 'x2'
salida(i,1)=2;
case 'x3'
salida(i,1)=3;
case 'x4'
salida(i,1)=4;
case 'x5'
salida(i,1)=5;
case 'x6'
salida(i,1)=6;
case 'x7'
salida(i,1)=7;
case 'x8'
salida(i,1)=8;
case 'x9'
salida(i,1)=9;
case 'x10'

```

```

        salida(i,1)=10;
        case 'x11'
            salida(i,1)=11;
    end
end
%%Porcentaje de componente principal
A2=0;
for i=2:1:10
    A1(1,i-1)=abs(table2array mdl.Coefficients(i,3));
    A2=A2+A1(1,i-1);
end
porcentaje=0;
for j=1:1:3
    A3=max(A1);
    porcentaje(j,1)=A3;
    for i=2:1:10
        if(A3==A1(1,i-1))
            A1(1,i-1)=0;
        end
    end
end
end
end

```

1. Adding x10, FStat = 5821.5587, pvalue = 0
2. Adding x5, FStat = 104.8405, pvalue = 1.830551e-23
3. Adding x2, FStat = 1011.9007, pvalue = 8.2908452e-154
4. Adding x2:x10, FStat = 435.1703, pvalue = 1.8701e-80
5. Adding x1, FStat = 284.6746, pvalue = 2.339482e-56
6. Adding x8, FStat = 1477.7145, pvalue = 9.1955446e-199
7. Adding x9, FStat = 214.8797, pvalue = 3.473473e-44
8. Adding x1:x10, FStat = 47.1207, pvalue = 1.17554e-11
9. Adding x2:x9, FStat = 97.7038, pvalue = 4.89366e-22
10. Adding x2:x8, FStat = 1428.0373, pvalue = 4.0658735e-194
11. Adding x1:x9, FStat = 721.1322, pvalue = 1.023271e-119
12. Adding x3, FStat = 451.2597, pvalue = 9.082867e-83
13. Adding x9:x10, FStat = 91.8305, pvalue = 7.39641e-21
14. Adding x7, FStat = 62.8962, pvalue = 5.88739e-15
15. Adding x2:x3, FStat = 21.9163, pvalue = 3.247e-06
16. Adding x1:x2, FStat = 71.436, pvalue = 1.02293e-16
17. Adding x3:x9, FStat = 57.3926, pvalue = 8.23047e-14
18. Adding x8:x10, FStat = 29.6759, pvalue = 6.45738e-08
19. Adding x4, FStat = 6.2847, pvalue = 0.012339
20. Adding x4:x10, FStat = 28.0258, pvalue = 1.4779e-07
21. Adding x3:x10, FStat = 84.5396, pvalue = 2.20926e-19
22. Adding x1:x4, FStat = 238.4058, pvalue = 2.728522e-48
23. Adding x3:x4, FStat = 290.877, pvalue = 2.768471e-57
24. Adding x2:x4, FStat = 2719.1411, pvalue = 2.8225097e-284
25. Adding x8:x9, FStat = 111.5294, pvalue = 9.303287e-25
26. Adding x1:x3, FStat = 5.6593, pvalue = 0.017555
27. Removing x5, FStat = 0.0060851, pvalue = 0.93784
28. Removing x8:x10, FStat = 0.19977, pvalue = 0.65501

Salida de solo las componentes principales

```

R_MRLM.CP=salida;
R_MRLM.PCP=porcentaje;

```



```
end
```

Normalización

```
function [Xn]=GPN(X)
% Normalización
aux=max(X);
[f c]=size(X);
Xn=[];
for i=1:1:f
    for j=1:1:c
        Xn(i,j)=X(i,j)/aux(j);
    end
end

end
```

Creación de Base de Datos

Base de datos sin procesamiento

```
function [BD]=GPBD()

load Caracteristicas
AX1=Caracteristica_Extension;
AX2=Caracteristica_Flexion;
AX3=Caracteristica_Pronacion;
AX4=Caracteristica_Supinacion;
X=[];
```

Entrenamiento

Elaboración de matriz de entrenamiento

```
for i=1.0:1.0:250.0
    for j=1:1:11
        X(i,j)=AX1(i,j);
        X((i+250),j)=AX2(i,j);
        X((i+500),j)=AX3(i,j);
        X((i+750),j)=AX4(i,j);
    end
end
% Normalización de matriz de entrenamiento
X=GPN(X);
BD.Entrenamiento=X;
```

Prueba

```
X=[];
% Elaboración de matriz de prueba
for i=1.0:1.0:250.0
    for j=1:1:11
        X(i,j)=AX1(i+250,j);
        X((i+250),j)=AX2(i+250,j);
```

```

        X((i+500),j)=AX3(i+250,j);
        X((i+750),j)=AX4(i+250,j);
    end
end
% Normalización de matriz de prueba
X=GPN(X);
BD.Prueba=X;
end

```

Base de datos procesada

```
function [BD]=GPBDP(BD,R_ACP,R_MRLM,Y)
```

ACP Entrenamiento

```

X=[];
for i=1:1:3
    X(:,i)=BD.Entrenamiento(:,R_ACP(i));
end
BD.R_PCA_Entrenamiento=X;

```

ACP Prueba

```

X=[];
for i=1:1:3
    X(:,i)=BD.Prueba(:,R_ACP(i));
end
BD.R_PCA_Prueba=X;

```

MRLM Entrenamiento

```

X=[];
for i=1:1:3
    X(:,i)=BD.Entrenamiento(:,R_MRLM(i));
end
BD.R_MRLM_Entrenamiento=X;

```

MRLM Prueba

```

X=[];
for i=1:1:3
    X(:,i)=BD.Prueba(:,R_MRLM(i));
end
BD.R_MRLM_Prueba=X;

```

Tipo de movimiento (Salida) String y Binario

```
X=[];
X1="";
[f c]=size(Y);
for i=1:1:f
switch Y(i)
    case 1
        X(i,:)= [1,0,0,0];
        X1(i,1)="Extension";
    case 2
        X(i,:)= [0,1,0,0];
        X1(i,1)="Flexion";
    case 3
        X(i,:)= [0,0,1,0];
        X1(i,1)="Pronacion";
    case 4
        X(i,:)= [0,0,0,1];
        X1(i,1)="Supinacion";
end
end
BD.S_RN=X;
BD.S_NS=table(X1);
end
```

Clasificación

Inicialización de datos

```
clear all, close all, clc
load BDGP
[~,~,y] = unique(BD.S_NS);
y       = [y; y];
%Si z=0 se usan las 11 dimensiones z=1 se usan 3 dimensiones
```

Clasificación de base de datos sin reducir dimensiones

```
X      = [BD.Entrenamiento; BD.Prueba];
z=0;
Clasificador(X,y,z)
```

Clasificación de base de datos resultado ACP

```
X      = [BD.R_PCA_Entrenamiento; BD.R_PCA_Prueba];
z=1;
Clasificador(X,y,z)
```

Clasificación de base de datos resultado MRLM

```
%X          = [BD.R_MRLM_Entrenamiento; BD.R_MRLM_Prueba];  
z=1;  
Clasificador(X,y,z)
```

Funcion de clasificacion

```
function Clasificador(X,y,z)  
if(z==0)  
A          = prdataset(X(:, [1:11]),y);  
end  
if(z==1)  
A          = prdataset(X(:, [1:3]),y);  
end  
  
Niter = 20;  
for i = 1:Niter  
    [Atrain,Atest] = gendat(A,0.8);  
    scatterd(Atrain)  
    scatterd(Atest)  
  
    wldc          = ldc(Atrain);  
    wqdc          = qdc(Atrain);  
    wknn          = knnc(Atrain);  
    wtotal        = {wldc, wqdc, wknn};  
    error_total(i,:) = testc(Atest*wtotal);  
    CP_total(i,:)   = 1-testc(Atest*wtotal);  
end  
  
boxplot(error_total)  
  
boxplot(CP_total)    %grafica  
  
end
```